

Inhaltsverzeichnis

Inhaltsverzeichnis	I
Abbildungsverzeichnis	III
Tabellenverzeichnis	V
Abkürzungsverzeichnis	VI
1 Zielsetzung und Motivation	1
2 RFID – Stand der Technik.....	3
2.1 <i>Physikalische und technische Grundlagen.....</i>	3
2.1.1 Frequenzbereiche RFID	3
2.1.2 Induktive Kopplung.....	4
2.1.3 Speicher in RFID-Transpondern.....	6
2.2 <i>Aufbau, Funktionsweise</i>	8
2.2.1 Aufbau Reader	8
2.2.2 Aufbau Transponder.....	9
2.2.3 Datenübertragung	10
2.2.4 Vielfachzugriff.....	15
2.3 <i>Unterscheidungen und Anwendungen</i>	19
2.3.1 Transponderarten.....	22
2.4 <i>Ortsbestimmung.....</i>	23
3 Versuchsaufbau	25
3.1 <i>RFID-System.....</i>	25
3.1.1 Schreib-/ Lesegerät	25
3.1.2 Antennen	27
3.1.3 Transponder.....	31
3.2 <i>Industrieroboter</i>	33

3.2.1	Greifer	34
3.3	<i>Gesamtsystem</i>	35
4	Durchführung	37
4.1	<i>Matlab</i>	37
4.1.1	Struktur der Datenpakete des RFID-Systems	38
4.1.2	Implementierte Funktionen des RFID-Systems	40
4.1.3	Sonstige implementierte Funktionen	44
4.2	<i>Ortung in der Ebene</i>	45
4.2.1	Sendeleistungskarte	46
4.2.2	Umsetzung in der Programmierung	48
4.3	<i>Intelligentes Greifen</i>	52
4.3.1	Hardware-Umsetzung	53
4.3.2	Software-Umsetzung	56
4.3.3	Eindimensionale Transponderortung	58
4.4	<i>Sortierung</i>	60
4.4.1	Sortieralgorithmen	60
4.4.2	Umsetzung der Sortierung	63
5	Fazit und Ausblick	66
	Literatur & Quellen	69
	Anlagen	I
	Anlagen, Teil 1	II
	Anlagen, Teil 2	IV
	Selbstständigkeitserklärung	79

Abbildungsverzeichnis

Abbildung 1 – Frequenzbereiche RFID	4
Abbildung 2 – Spannungsversorgung über induktive Kopplung	6
Abbildung 3 – Ersatzschaltbild Reader	9
Abbildung 4 – Manchester-Codierung	12
Abbildung 5 – Pulsphasenmodulation	14
Abbildung 6 – S-ALOHA-Verfahren.....	18
Abbildung 7 – Anwendungsbereiche RFID	21
Abbildung 8 – Smart-Label.....	22
Abbildung 9 – Ortsbestimmung mit RFID	24
Abbildung 10 – Reader	25
Abbildung 11 – Kommunikationszeit in Abhängigkeit der Transponderanzahl.....	27
Abbildung 12 – Lesereichweite zweier Transponder in Abh. der Sendeleistung.....	28
Abbildung 13 – magnetische Feldlinien der Antenne	29
Abbildung 14 – Platinen-Antennen.....	30
Abbildung 15 – Transponder	32
Abbildung 16 – IRB 120	33
Abbildung 17 – Greifer WSG 50.....	35
Abbildung 18 – Gesamtsystem	36
Abbildung 19 – Orientierung	46

Abbildung 20 – Karte	48
Abbildung 21 – Struktogramm Ortung über Sendeleistung.....	48
Abbildung 22 – RFID-Antenne an Industrieroboter	50
Abbildung 23 – Raster & Messpunkte.....	51
Abbildung 24 – Antennen an Greifer.....	53
Abbildung 25 – RFID-Empfangsbereich mit ID ISC.ANT100/100 (Maße in mm) ..	55
Abbildung 26 – Bausteine mit integrierten Transpondern	56
Abbildung 27 – Eindimensionale Ortung.....	59
Abbildung 28 – Bildfolge Sortieren.....	65

Tabellenverzeichnis

Tabelle 1 – Speicher in RFID-Transpondern.....	7
Tabelle 2 – Multiplexverfahren	16
Tabelle 3 – Chipkartenarten	20
Tabelle 4 – Struktur der Datenpakete	38
Tabelle 5 – Implementierte Funktionen Roboter und Greifer	45
Tabelle 6 – Bubblesort und Selectionsort.....	61
Tabelle 7 – Beispiel Sortieralgorithmen.....	62

Abkürzungsverzeichnis

ASCII	American Standard Code for Information Interchange
CDMA	Code-Division Multiple Access
CRC	Cyclic Redundancy Check
EEPROM	Electrically Erasable Programmable Read Only Memory
FRAM	Ferroelectric Random Access Memory
FDMA	Frequency-Division Multiple Access
HF	High Frequency
ISO	International Organization for Standardization
IEC	International Electrotechnical Commission
LF	Low Frequency
MW	Medium Frequency
RAM	Random-Access Memory
RFID	Radio Frequency Identification
ROM	Read Only Memory
SDMA	Space-Division Multiple Access
TCP/IP	Transmission Control Protocol/Internet Protocol
TDMA	Time-Division Multiple Access
UHF	Ultra High Frequency
USB	Universal Serial Bus

1 Zielsetzung und Motivation

Die Bedeutung von Identifikationssystemen nimmt in Technik und Wirtschaft ständig zu. Man unterscheidet nach der Art der Informationsübertragung in optische, mechanische und elektromagnetische Systeme. Mich hat das als Masterstudent stark interessiert. Deshalb habe ich mich intensiv mit Radio Frequency Identification (RFID) beschäftigt, um im Ergebnis künftigen Studenten der Hochschule Mittweida dieses Thema näher zu bringen.

Identifikationssysteme dienen der Speicherung von Informationen auf Objekten oder Lebewesen und somit zur Unterscheidung dieser. Ein weiterer Vertreter dieser Systeme ist der Barcode. Diese weit verbreitete Technologie hat jedoch einige Nachteile gegenüber der RFID-Technik: Man kann Barcodes nicht verändern. Die Informationen sind nicht im Barcode selbst gespeichert, sie verweisen lediglich auf eine übergeordnete Datenbank. Außerdem werden die Barcodes optisch – nicht elektromagnetisch – ausgelesen. Das bedeutet, dass eine Sichtverbindung zwischen Lesegerät und Objekt bestehen muss, der Barcode nicht verschmutzt sein darf und stets nur ein Objekt gleichzeitig ausgelesen werden kann. Jede einzelne dieser Tatsachen ist ein Argument für den Einsatz von RFID-Systemen an Stelle von Barcodes in industriellen Umgebungen.¹ Aus diesem Grund soll in dieser Arbeit sowohl auf die Eigenschaften von RFID-Systemen, als auch auf mögliche Anwendungen dieser Technologie in der Industrie eingegangen werden.

Obwohl die Grundlagen für die RFID-Technologie bereits Ende des zweiten Weltkrieges geschaffen wurden², ist die Prosperität dieser Technologie seit einigen Jahren stärker als je zuvor. Jeder kommt nahezu täglich mit RFID-Technologie in Kontakt. Die Einsatzbereiche reichen von simplen Anwendungen, wie der Diebstahlsicherung in Kaufhäusern, über die Tieridentifikation bis hin zu komplexen

¹ [Fink2016], S.2ff

² [Lamm2017]

Anwendungen in der Industrie. Die wohl aktuellste Verwendung von RFID-Systemen bilden Chipkarten zur bargeldlosen Bezahlung oder sonstigen Personenidentifikation. In dieser Masterarbeit werden die industriellen Einsatzbereiche, besonders aus der industriellen Automation, von RFID im Vordergrund stehen. Dazu soll ein RFID-System mit einem Industrieroboter und einem daran montierten Greifer kombiniert werden. Diese Teilsysteme lagen bereits aus vorangegangenen Projekten an der Hochschule vor und sollten um die RFID-Funktionen erweitert werden.

Diese Masterarbeit, sowie die damit verbundenen Versuche und Programmierungen, hat das Ziel, die Grundlage für Praktika künftiger Studenten der Hochschule Mittweida zu bilden. Auf diese Weise kann den Studenten künftig das Thema RFID auf praktische Weise vermittelt und ein Einblick in das breite Spektrum der möglichen Anwendungen von RFID gegeben werden.

2 RFID – Stand der Technik

Die Abkürzung RFID steht für Radio Frequency Identification. Frei übersetzt handelt es sich dabei um kontaktlose Identifikationssysteme. Diese haben die Aufgabe, Informationen zum Produkt, auf dem sie sich befinden, bereitzustellen.

Im Gegensatz zu anderen Identifikationssystemen, ist die Information im RFID-Transponder direkt gespeichert. Somit sind RFID-Systeme autark einsetzbar – es ist kein Computer / keine Datenbank zur Verwendung nötig. Die Informationen werden vom Lesegerät mit Hilfe von Elektromagnetischen Wellen kontaktlos ausgelesen und gegebenenfalls verändert. Genauer zu der Funktionsweise, sowie zu verschiedenen RFID – Systemen und deren Anwendungen werden in diesem Kapitel behandelt.³

2.1 Physikalische und technische Grundlagen

Bevor im Abschnitt 2.2 auf die Funktionsweise von RFID-Systemen eingegangen wird, sollen in diesem Kapitel die physikalischen Grundlagen beleuchtet werden.

2.1.1 Frequenzbereiche RFID

Die Informationsübertragung in RFID basiert auf Radiowellen. Die dabei verwendeten Frequenzbänder reichen vom Langwellenfrequenzbereich (9 kHz - 135 kHz) bis in den Mikrowellenbereich (2,45 GHz beziehungsweise 5,8 GHz). Selbstverständlich arbeiten noch zahlreiche andere Technologien in diesem breiten Frequenzspektrum. Aus diesem Grund haben sich einige Frequenzen im heutigen Gebrauch etabliert. Diese sind in Abbildung 1 dargestellt. Wie ersichtlich, sind von diesen üblichen Frequenzen nicht alle standardisiert. Durch die vielfältigen Anwendungsmöglichkeiten von RFID existieren zahlreiche Normen. Die Norm für die

³ [Lamm2017]

Verwaltung von Gegenständen und somit die für diese Masterarbeit relevante Norm ist die ISO/IEC 18000. Darin sind für jede standardisierte Frequenz (orange in Abbildung 1) gesonderte Parameter für die kontaktlose Kommunikation festgelegt. Dadurch ergeben sich für jedes Frequenzband Vorteile und Nachteile. Diese ergeben sich aus den physikalischen Zusammenhängen der Parameter mit der Frequenz.⁴

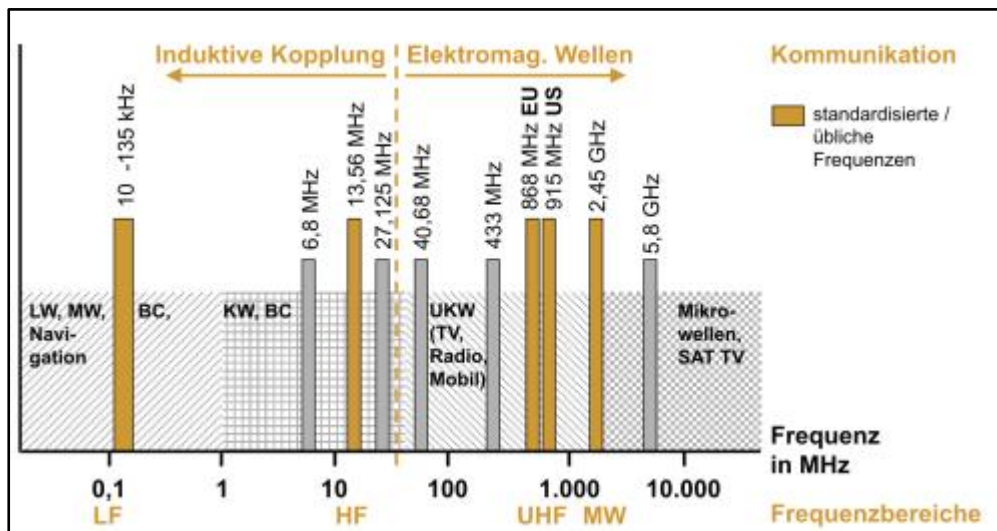


Abbildung 1 – Frequenzbereiche RFID⁵

2.1.2 Induktive Kopplung

Wie in Abbildung 1 dargestellt ist, ist es bei Wahl einer Betriebsfrequenz aus dem Hochfrequenz- (HF) oder Niedrigfrequenzbereich (LF) möglich, die Spannungsversorgung⁶ des Transponders über induktive Kopplung zu realisieren. In Frequenzbereichen oberhalb von 30 MHz (gestrichelte Linie in Abbildung 1) ist es üblich, dass der Transponder eine eigene Spannungsversorgung hat.

Die induktive Kopplung hängt in erster Linie vom Abstand zwischen Transponder und Schreib-/ Leseinheit ab. Solange sich der Transponder im Nahfeld der Antenne des Readers befindet, ist die Spannungsversorgung über induktive Kopp-

⁴ [Shar2008]

⁵ [OvVo2005], S.2

⁶ [OvVo2005], S.2f, [Koln2001], S.10f

lung theoretisch möglich. Man spricht dann von einem passiven Transponder. Das Nahfeld einer Antenne ist der Bereich um die Antenne in dem folgendes gilt: $Abstand < Wellenlänge / 2\pi$. Da die Wellenlänge über den Zusammenhang $Wellenlänge = Lichtgeschwindigkeit / Frequenz$ definiert ist, ist die Induktive Kopplung frequenzabhängig. Mit steigender Frequenz wird die Reichweite von induktiver Kopplung geringer. Deshalb muss der Transponder bei hohen Frequenzen (UHF, MW) eine eigene Spannungsversorgung – in der Regel eine Batterie – aufweisen. In diesem Fall spricht man von einem aktiven Transponder. Mehr über aktive und passive Transponder ist unter Punkt 2.3.1 zu finden.⁷

Es müssen noch weitere Voraussetzungen erfüllt sein, damit die Spannungsversorgung über Induktive Kopplung erfolgen kann. Der maximale Bereich für induktive Kopplung um die Antenne des Readers hängt des Weiteren von der Sendeleistung, der Größe der Sendeantenne, der Größe der Transponderantenne und der Beschaffenheit des Raumes zwischen den Antennen ab.⁸

Da die Betriebsfrequenz des RFID-Systems, das im Rahmen der Versuche dieser Masterarbeit verwendet wurde, mit 13,56 MHz im HF-Bereich liegt und somit die Kommunikation zwischen Reader und Transponder sowie deren Energieversorgung auf induktiver Kopplung basiert, soll sich in dieser Arbeit auf diese Vorgänge konzentriert werden.

Die induktive Kopplung oder auch Gegeninduktion ist ein physikalisches Phänomen. Man spricht davon, wenn sich in der näheren Umgebung zu einer stromdurchflossenen Leiterschleife eine weitere Leiterschleife befindet und so in der zweiten Leiterschleife eine Spannung induziert wird. Dieses Prinzip nutzt man nicht nur bei Transformatoren, sondern auch bei der Spannungsversorgung des passiven RFID-Transponders. Die Schreib-/ Leseinheit beinhaltet die stromdurchflossene Leiterschleife, welche im sekundären Stromkreis (im Transponder) - sobald er in den Bereich der induktiven Kopplung gelangt – eine Spannung induziert. Diese Spannung muss im Transponder lediglich noch gleichgerichtet werden

⁷ [OvVo2005], S.2f, [Koln2001], S.10f

⁸ [OvVo2005], S.2f, [Koln2001], S.10f

und dient so der Spannungsversorgung des Mikrochips.⁹ Das Prinzip der Spannungsversorgung des Transponders über induktive Kopplung und die dazu nötigen Schaltungen in Reader und Transponder sind in Abbildung 2 dargestellt.

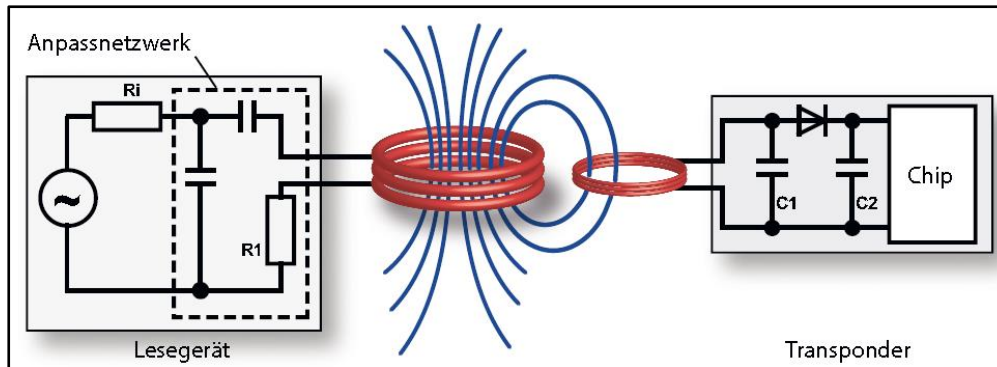


Abbildung 2 – Spannungsversorgung über induktive Kopplung¹⁰

Die blauen Linien in Abbildung 2 bezeichnet man als magnetische Feldlinien. Ihr Verlauf in dieser Darstellung ist typisch für eine stromdurchflossene Leiterschleife, welche die Antenne von Reader und Transponder bildet. Magnetische Feldlinien sind im Gegensatz zu elektrischen Feldlinien stets geschlossen.

2.1.3 Speicher in RFID-Transpondern

Wie bereits unter Punkt 2 erwähnt, können in RFID-Transpondern direkt Informationen gespeichert werden. Ein Transponder muss deshalb eine eigene, integrierte Speichereinheit besitzen und nicht nur auf eine Datenbank eines übergeordneten Computersystems verweisen. Die Speichereinheiten verschiedener Transponderarten unterscheiden sich dabei allerdings in der maximalen speicherbaren Datenmenge und in der Art der Speicherrealisierung. Es existieren zahlreiche verschiedene Arten der Speicherrealisierung. So unterteilt man grundsätzlich in Speicher, die nur gelesen werden können (engl. read only memory – ROM) und Speicher, die vom Anwender programmiert werden können. Wenn man einige weitere Un-

⁹ [Fink2016], S.15

¹⁰ [Fink2016], S.15

terscheidungen vornimmt, erhält man die beiden Speichertypen, die heutzutage vorrangig – für induktiv gekoppelte Transponder – eingesetzt werden. Diese wurden in Tabelle 1 gegenübergestellt.¹¹

Name	Bedeutung	Prinzip
EEPROM	electrically erasable programmable read only memory	<ul style="list-style-type: none"> • 1 Bit pro Speicherzelle (= 1 Schalttransistor & 1 Speichertransistor) • Datenspeicherung durch Ladungsspeicherung im isolierten Gate des Speichertransistors • Je nach Ladung 1 oder 0
FRAM	ferromagnetic random access memory	<ul style="list-style-type: none"> • 1 Bit pro Speicherzelle (= 1 ferroelektrischer Kondensator & 1 Transistor) • Polarisation von ferroelektrischen Kristallen im Dielektrikum der Kondensatoren • Je nach Polarisation 1 oder 0

Tabelle 1 – Speicher in RFID-Transpondern¹²

Diese beiden Speichertypen eignen sich für die Verwendung in induktiv gekoppelten RFID-Transpondern, weil sie nicht flüchtig sind. Das bedeutet, dass sie auch ohne permanente Spannungsversorgung die Daten im Speicher behalten. Da der Schwerpunkt dieser Masterarbeit nicht im Bereich der Datenspeicher liegt, wird auf die weitere Erläuterung der Speichertypen verzichtet.

¹¹ [Stin2015], S.569f; [Fink2016], S.14

¹² [Fink2016], S.14, [Stin2015], S.586ff, S.600ff

2.2 Aufbau, Funktionsweise

Ein RFID – System besteht stets aus einer Schreib-/ Leseinheit (meist Reader genannt) und mindestens einem Transponder. Die Schreib-/ Leseinheit kann über verschiedene Schnittstellen an ein übergeordnetes Rechensystem angeschlossen werden. Auch der Anschluss von mehreren Schreib-/ Leseinheiten an das Rechensystem ist möglich.

2.2.1 Aufbau Reader

Abbildung 3 zeigt ein prinzipielles Ersatzschaltbild eines Schreib-/ Lesegerätes. Dabei handelt es sich lediglich um die Bauteile, die das Gerät mindestens aufweisen muss. Die Induktivität L_1 und der Widerstand R_1 stehen hier für die Antenne des Readers. Wobei R_1 für den ohmschen Verlust der Antennenleiterschleife steht. Zusammen mit dem Kondensator C_1 bilden die Bauteile der Antenne einen Serienresonanzkreis. Die komplexe Gesamtimpedanz $|Z|$ eines Serienresonanzkreises¹³ ist frequenzabhängig und berechnet sich folgendermaßen:

$$|Z| = \sqrt{R^2 + \left(\omega L - \frac{1}{\omega C}\right)^2}$$

Darin steht ω für die Kreisfrequenz:

$$\omega = 2\pi f$$

Die Gesamtimpedanz wird minimal, wenn der Klammerterm in der Gleichung der Gesamtimpedanz Null wird. Dann wirkt nur noch der ohmsche Widerstand. Dies passiert bei folgender Frequenz:

$$f_r = \frac{1}{2\pi\sqrt{LC}}$$

In diesem Fall spricht man von Resonanz und der Resonanzfrequenz f_r .

¹³ [Prec2007], S.210

Wenn die Betriebsfrequenz des Readers der Resonanzfrequenz des Serienresonanzkreises aus $R1$, $L1$ und $C1$ in Abbildung 3 entspricht, wird gewährleistet, dass die Gesamtimpedanz $|Z|$ minimal wird und somit der Stromfluss durch die Antenne maximal ist.¹⁴

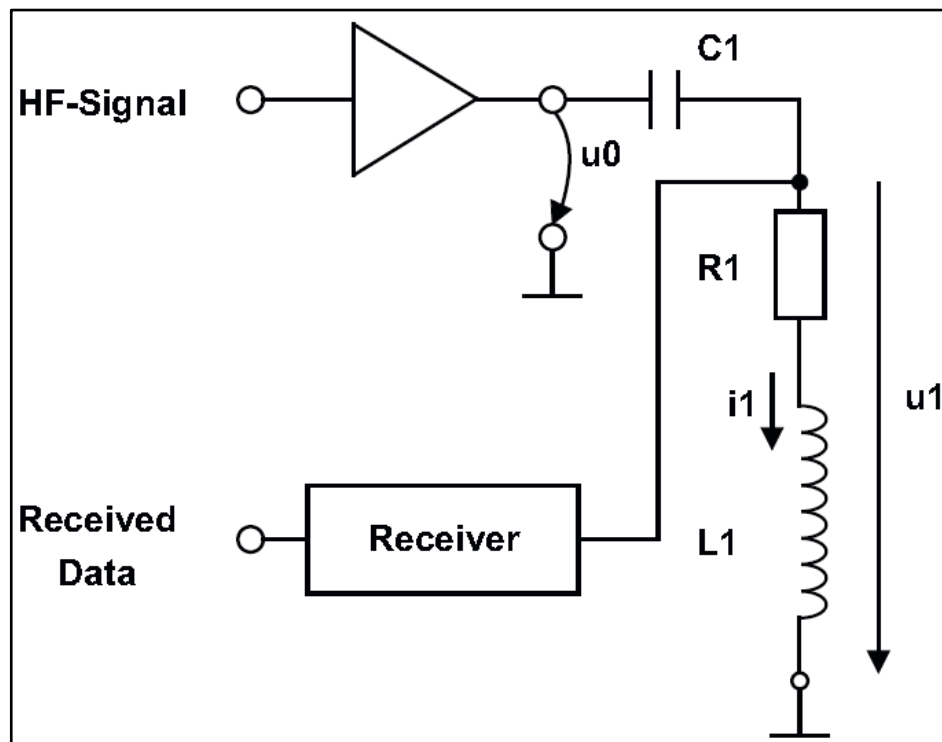


Abbildung 3 – Ersatzschaltbild Reader¹⁵

2.2.2 Aufbau Transponder

Das Wort Transponder setzt sich aus den Wörtern Transmitter und Responder zusammen. Wörtlich übersetzt ist ein Transponder also ein Antwortsendegerät. Die Schaltung eines Transponders ist vergleichsweise einfach und ist je nach Anwendung in einem Gehäuse, einer Chipkarte oder auf einem selbstklebenden Träger untergebracht. Der Aufbau und die Funktionsweise des Transponders unterscheiden sich je nach Anwendung.

¹⁴ [Prec2007], S.210

¹⁵ [Fink2016], S.104

So besteht ein 1-Bit-Transponder¹⁶ lediglich aus einer Spule, die als Antenne fungiert und einem in Reihe geschalteten Kondensator. Diese einfache Schaltung bildet somit einen LC-Schwingkreis, dessen Resonanzfrequenz genau auf den Betriebsfrequenzbereich des Readers abgestimmt ist. Sobald der 1-Bit-Transponder in den Bereich der induktiven Kopplung des Readers kommt, wird der Schwingkreis zur Resonanzschwingung angeregt. Wie unter Punkt 2.1 beschrieben, wird durch den entstehenden Stromfluss im Transponder dem Magnetfeld des Readers Energie entzogen. Dies wird vom Reader registriert und der Reader erkennt, dass sich ein Transponder im Einflussbereich befindet. Dieses Verfahren, die Induktive Kopplung, ist nur eines von mehreren - allerdings das wichtigste und am häufigsten verwendete - zur Detektion von 1-Bit-Transpondern.

Die Schaltung aller anderen Transponderarten, also der Transponder, die Teil eines RFID-Systems sind, das mit sequentiellen Verfahren oder mit Halb-/ Voll-duplexverfahren arbeitet, beinhaltet einen Mikrochip. Dieser fungiert als Datenträger von Datenmengen von wenigen Bytes bis hin zu einigen Megabytes. Auch die Beschreibung dieser Transponderart soll in dieser Masterarbeit auf die Induktive Kopplung beschränkt werden, weil auch das im Versuchsaufbau verwendete RFID-System auf dieser Kommunikationsweise basiert. Generell kann man sagen, dass die Transponder, die über induktive Kopplung kommunizieren, fast ausschließlich passive Transponder sind.¹⁷ Die integrierte Spule hat also die Funktion der Spannungsversorgung des Transponders via Gegeninduktion und ist für den Datentransport vom Reader zum Transponder und entgegengesetzt verantwortlich.

2.2.3 Datenübertragung

Ebenso wie die Energieversorgung der Transponder kann auch die Datenübertragung zwischen Transponder und Reader auf verschiedene Weisen erfolgen. Für nahezu jeden Frequenzbereich, jede Kopplungsart und jeden Transpondertyp gibt

¹⁶ [RFID2016]; [Schn2016]; [Fink2016], S.45

¹⁷ [RFID2016]; [Fink2016], S.47

es spezielle Übertragungsverfahren. An dieser Stelle soll allerdings lediglich auf die Datenübertragung im Frequenzbereich des verwendeten RFID-Systems (13,56 MHz) bei induktiver Kopplung mit den in Abschnitt 3.1.3 erläuterten Transpondern eingegangen werden. Selbstverständlich existiert für jeden Fall eine Norm, in der unter anderem die Form der Datenübertragung geregelt ist.

Der Datentransfer im verwendeten RFID-System ist in der ISO15693¹⁸ geregelt. Diese Norm ist auch Bestandteil der bereits unter Punkt 2.1 erwähnten ISO 18000. Als Downlink bezeichnet man die Kommunikation vom Transponder zum Reader, der Uplink ist demnach der Datentransfer in die entgegengesetzte Richtung. Beide funktionieren auf eine andere Weise, auf welche an dieser Stelle eingegangen werden soll.

Der Downlink von Transponder zu induktiv gekoppeltem RFID-Reader im Halb- oder Vollduplexbetrieb basiert auf der Lastmodulation¹⁹ mit Hilfsträger. Sie funktioniert, wie die unter Punkt 2.1 erläuterte Spannungsversorgung, lediglich im Nahfeld des Readers, der Transponder darf also nicht weiter als das Ergebnis der Gleichung $Wellenlänge / 2\pi$ vom Schreib-/ Lesegerät entfernt sein. Befindet sich ein Transponder in diesem Gebiet und wird wie oben beschrieben vom Feld, das vom Lesegerät erzeugt wird, mit Energie versorgt, findet die Kommunikation über die Veränderung der Energieentnahme aus dem Magnetfeld statt.

Um genauer auf den Vorgang der Datenübertragung vom Transponder zum Reader einzugehen, müssen zuerst die Begriffe Signalcodierung und Signalmodulation definiert werden. Beide Begriffe sind im RFID-Handbuch²⁰ erläutert: „Eine Signalcodierung hat die Aufgabe, eine zu übertragende Nachricht und ihre Signaldarstellung möglichst optimal an die Eigenschaften des Übertragungskanals anzupassen. Dazu gehört die mehr oder weniger gute Sicherung der Nachricht gegen Störungen oder Kollision und die gezielte Veränderung bestimmter Signaleigenschaften. [...] Modulation ist die Veränderung von Signalparametern eines hoch-

¹⁸ [ISO2006]

¹⁹ [RFID2016]; [Fink2016], S.51

²⁰ [Fink2016], S.219

frequenten Trägers, also Amplitude, Frequenz oder Phase, in Abhängigkeit von einem modulierenden Signal, dem Basisbandsignal.“²¹

Das zu übertragende Signal liegt im Transponder in binärer Form vor. Dieses Signal wird im Mikrochip des Transponders mit dem Manchester-Code²² codiert. Abbildung 4 zeigt das Prinzip dieser Codierung, die an der Universität Manchester entwickelt wurde. Dabei wird das zu übertragende, binäre Signal mit dem sogenannten Trägersignal (Takt in Abbildung 4), per Exklusiv-Oder (XOR) verknüpft. Wobei die Frequenz des Trägersignals mit 450 kHz doppelt so hoch ist wie die des Datensignals. Das Ergebnis der Codierung ist ein binäres Signal, mit einer logischen 1 an den Stellen mit einer positiven Flanke des Trägersignals in der Mitte des Datensignals und einer logischen 0 an den Stellen mit einer negativen Flanke des Trägersignals in der Mitte des Datensignals.

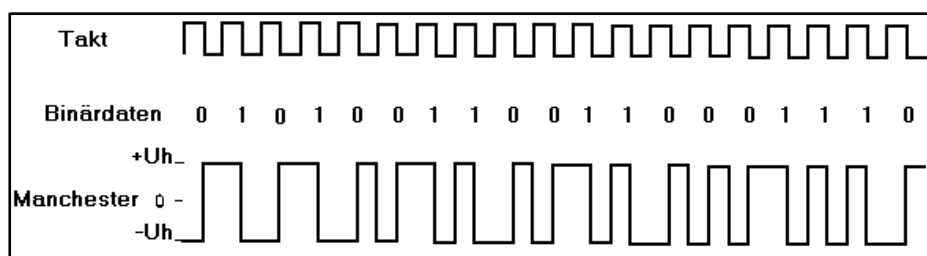


Abbildung 4 – Manchester-Codierung²³

In Abhängigkeit dieses codierten Signals wird nun ein Lastwiderstand im Transponder zugeschaltet. Diese Modulationsart heißt Amplitudenmodulation²⁴ mit Hilfsträger oder On-Off-Keying. Der Hilfsträger ist in diesem Fall das zuvor codierte Signal mit der Frequenz von 450 kHz. Durch das Zuschalten des Lastwiderstandes verringert sich die Amplitude des hochfrequenten (13,56 MHz) magnetischen Feldes des Readers an den Stellen, wo das codierte Signal, beziehungsweise der Hilfsträger einen High-Zustand (logische 1) hat.

²¹ [Fink2016], S.219

²² [Bral2017]

²³ [Bral2017]

²⁴ [Fink2016], S.223, S.379]

Die Energieaufnahme beziehungsweise die Last des Transponders erhöht/verringert sich also in Abhängigkeit des zu übertragenden Signals. Diese Lastveränderungen rufen durch die induktive Kopplung eine Veränderung der Impedanz der Antennenspule des Readers hervor.²⁵ Der Transponder moduliert also die Spannung, die durch die Antennenspule des Readers fließt. Aus diesem Grund spricht man von Lastmodulation. Nun wird diese Spannung mit Hilfe einer Diode im Reader demoduliert und das binäre Modulationssignal kann nach Glättung mittels Tiefpass im Mikrochip des Readers zurückgewonnen werden. Somit ist die Datenübertragung vom Transponder zum Schreib-/ Lesegerät abgeschlossen. Mehr zu diesem Thema kann man im RFID-Handbuch von Klaus Finkenzeller in den Kapiteln 3.2.1.2 und 6.2.4 nachlesen. Details über die physikalischen Grundlagen der Lastmodulation sind der Literatur²⁶ zu entnehmen.

Die Datenübertragung von der Schreib-/ Leseinheit zum Transponder funktioniert auf andere Art und Weise. So basiert die Kommunikation in dieser Richtung nach ISO 15693 auf der Pulsphasenmodulation als Codier-Verfahren und erneut einer Art von Amplitudenumtastung als Modulationsverfahren. Von beiden Verfahren sind jeweils zwei Varianten möglich, sodass die Kommunikation insgesamt auf vier verschiedenen Kombinationen aus Codierungs- und Modulationsverfahren basieren kann. Während ein Transponder nach ISO 15693 alle vier Kombinationen beherrschen muss, müssen Lesegeräte dies nicht.²⁷

²⁵ [WiME2006], S.65

²⁶ [WiME2006], S.65

²⁷ [Fink2016], S.379ff

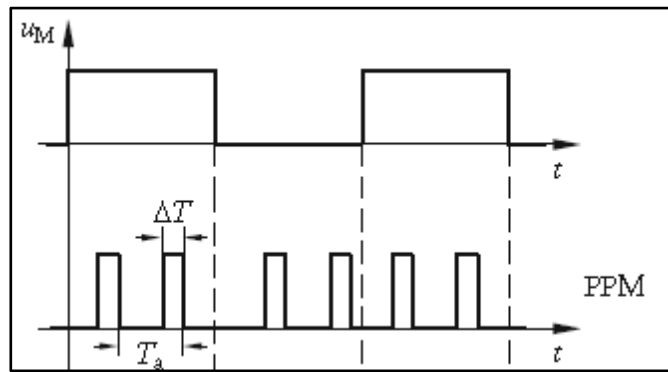


Abbildung 5 – Pulsphasenmodulation²⁸

Die beiden Codierungsverfahren sind Formen der Pulsphasenmodulation. Bei der Pulsphasenmodulation wird, wie Abbildung 5 zeigt, das zu übertragende Datensignal in einer Phasenverschiebung in Bezug zum Trägersignal codiert. Solange ein High-Signal (logische 1) des Datensignals anliegt, ist das codierte Signal eine regelmäßige Pulsfolge mit der Frequenz des Trägersignals. Die Frequenz des Trägersignals – Trägerfrequenz = $1 / \text{Periodendauer } T_a$ in Abbildung 5 – wird durch das Abtasttheorem von Shannon bestimmt. Das bedeutet, dass die Trägerfrequenz mindestens doppelt so hoch wie die Frequenz des Datensignals ist. Erst wenn ein Low-Signal (logische 0) des Datensignals anliegt, werden die Pulse des Trägersignals um eine bestimmte Phase verschoben. Diese Verschiebung bleibt bestehen bis wieder ein High-Signal anliegt.²⁹ Bei der ersten, langsameren Codierung, der „1 aus 256 Codierung“³⁰, werden 256 gleichlange Zeitschlitz pro Byte übertragen. Nur einer dieser Zeitschlitz wird dabei moduliert, beziehungsweise in nur einem Zeitschlitz werden die Impulse phasenverschoben übertragen. Welcher Zeitschlitz moduliert wird, entscheidet das zu übertragende Byte – mit einem Byte sind bekanntlich 256 verschiedene dezimale Zahlen (0 bis 255) darstellbar. Je nach Wert und Position der einzelnen Bits des Bytes wird also der entsprechende Zeitschlitz moduliert. Das zweite, schnellere, Codierungs-Verfahren funktioniert ganz ähnlich. Bei der „1 aus 4 Codierung“ besteht ein Zeitfenster jedoch nur aus 4 Zeitschlitz, von welchen wieder nur einer moduliert wird. Pro Zeitfenster (4 Zeit-

²⁸ Vgl. [PISc2016], S.1222

²⁹ [PISc2016], S.1222

³⁰ [Fink2016], S.380

schlitzen) können demnach 2 Bit übertragen werden. Ein zu überragendes Byte wird also mit viermal 2 Bit codiert.

Im Gegensatz zur Codierung ähnelt die Modulierung des Sendesignals des Readers der Modulierung des Sendesignals des Transponders. Die beiden Arten der Amplitudenmodulation des Sendesignals im Reader unterscheiden sich dadurch, dass bei der „100 % Amplitudenmodulation“ die Aussendung des magnetischen Feldes bei Anlegen eines LOW-Signals im codierten Datensignal komplett unterdrückt wird. Im Fall der Anwendung der „10% Amplitudenmodulation“ wird das ausgesendete Signal / das magnetische Feld des Readers bei der Übertragung einer logischen 0 lediglich um 10 bis 30 Prozent gedämpft. Da dieses Signal – das magnetische Feld des Readers mit der Frequenz von 13,56 MHz – auch zur Energieversorgung der Transponder genutzt wird, müssen die Transponder die kurzzeitige Dämpfung oder den kurzzeitigen Ausfall der Spannungsversorgung kompensieren können. In jedem Fall kann der Transponder, wenn er sich im Empfangsbereich des Readers befindet und mit Energie versorgt wird, das übertragene Signal demodulieren, interpretieren und anschließend auf dem soeben erklärtem Weg antworten.³¹

2.2.4 Vielfachzugriff

Von Pulk-Erfassung oder Vielfachzugriffsverfahren spricht man, wenn mehrere Transponder gleichzeitig an ein einzelnes Lesegerät senden wollen. Die Herausforderung ist in diesem Fall die Unterscheidung der Signale der verschiedenen Transponder. Es darf nicht zu Überlagerungen oder Kollisionen der Signale von Transpondern zum Lesegerät kommen, denn dann werden die Signale unbrauchbar. Die korrekte Signalunterscheidung kann gewährleistet werden indem die Datenströme der Transponder im Empfangsbereich unterschieden werden. Ferner kommen sogenannte Antikollisionsverfahren oder Multiplexverfahren zum Einsatz. Für RFID-Systeme unterscheidet man vier grundsätzliche Multiplexverfahren, die in Tabelle 2 aufgeführt sind. Wobei Kombinationen aus diesen Grundformen eben-

³¹ [Fink2016], S.379ff

falls möglich sind. Jedes dieses Verfahren unterscheidet die Transponder hinsichtlich genau einer Eigenschaft, nach dieser sie auch benannt sind.³²

Name	Beschreibung
Frequenzmultiplexverfahren (FDMA)	Transponder antworten gleichzeitig auf unterschiedlichen Trägerfrequenzen
Zeitmultiplexverfahren (TDMA)	Jeder Transponder im Empfangsbereich hat ein eigenes Zeitfenster in dem nur er kommuniziert
Raummultiplexverfahren (SDMA)	Räumliche Unterscheidung der Transponder durch Veränderung des Lesebereichs der (Richt-) Antenne oder durch Bewegung der Transponder (z.B. auf Fließband)
Codemultiplexverfahren (CDMA)	Analyse der Transpondersignale, vorübergehendes Abschalten ungewollter Transponder anhand dieser Analyse bis nur noch ein Transponder übrig bleibt

Tabelle 2 – Multiplexverfahren³³

Sehr verbreitet in RFID-Systemen ist die Anwendung von Zeitmultiplexverfahren. Es existieren zahlreiche verschiedene zeitliche Unterscheidungsverfahren. Zwei davon sollen hier kurz erläutert werden – das ALOHA-Verfahren und das Slotted-ALOHA-Verfahren (S-ALOHA). Diese Verfahren haben ihren Ursprung auf Hawaii

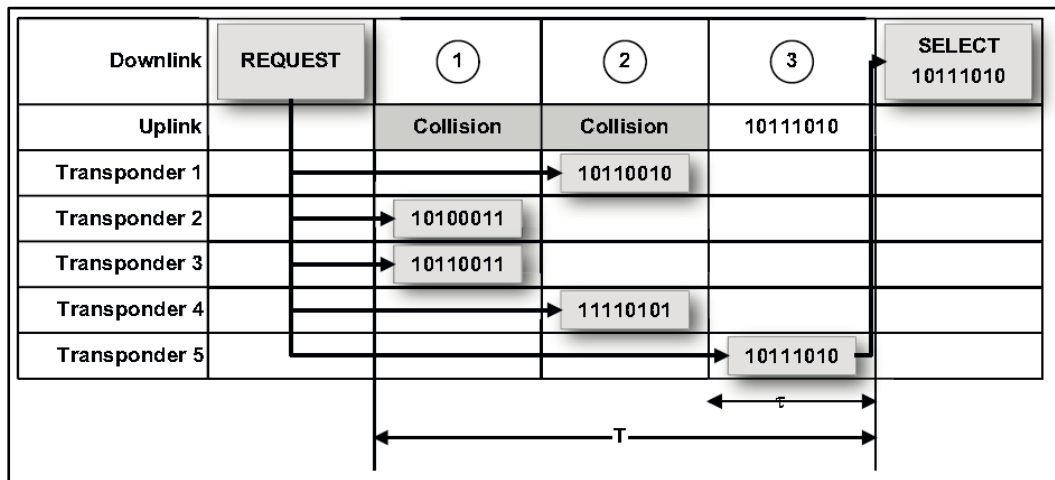
³² [Kern2011], S.34ff; [Fink2016], S.251ff

³³ [Kern2011], S.34ff; [Fink2016], S.251ff

– deshalb tragen sie den Name „ALOHA“. Im ALOHA-Verfahren³⁴ senden die Transponder im Empfangsbereich in zufälligen Zeitabständen ihre Seriennummer ein – im Vergleich zur (zufällig langen) Sendepause – sehr kurzes Signal. Da die Sendepausen der verschiedenen Transponder unterschiedlich lang sind, ist es nur eine Frage der Zeit bis eine Seriennummer kollisionsfrei an den Reader übertragen wurde. Kollisionen werden, wie in den Abschnitten 4.1.1 und 4.1.2 erklärt, vom Reader durch CRC-Prüfsummenprüfung erkannt. Im Anschluss hat der Reader die Möglichkeit, unter Verwendung der Seriennummer direkt mit diesem Transponder zu kommunizieren. Nach Abschluss dieser Kommunikation wird das Verfahren erneut gestartet, um den nächsten Transponder zu adressieren. Es ist allerdings möglich, dass erneut die gleiche Seriennummer zuerst kollisionsfrei übertragen wurde. Dieses Problem wird bei RFID-Systemen nach ISO15693, wie weiter unten in diesem Abschnitt erläutert, umgangen. Das S-ALOHA-Verfahren³⁵ unterscheidet sich dadurch, dass vom Reader Zeitfenster vordefiniert werden. Die Transponder wählen lediglich zufällig eines dieser Zeitfenster um ihre Seriennummer zu übertragen. Ansonsten gleicht das S-ALOHA-Verfahren dem ALOHA-Verfahren. Abbildung 6 verdeutlicht die Transponderauswahl im S-ALOHA-Verfahren. Die Spalten der Tabelle in dieser Abbildung stehen dabei für die vom Reader vorgegebenen Zeitfenster.

³⁴ [Kern2011], S.35; [Fink2016], S.257

³⁵ [Kern2011], S.35; [Fink2016], S.259

Abbildung 6 – S-ALOHA-Verfahren³⁶

Das in dieser Masterarbeit verwendete RFID-System arbeitet, wie bereits erwähnt, nach ISO15693. Das darin beschriebene Antikollisions-Verfahren arbeitet mit einer Kombination aus dem Zeitmultiplexverfahren und dem Codemultiplexverfahren.³⁷ Ein Reader, der Codemultiplexverfahren beherrscht, ist in der Lage, die Transponder im Empfangsbereich in drei verschiedene Modi zu schalten – Ready (Bereit), Quiet (Still) und Selected (Ausgewählt). Zu Beginn der Kommunikation sendet der Reader ein Signal an alle Transponder im Empfangsbereich – ein sogenanntes Broadcast-Signal. Inhalt dieses Signal ist ein Befehl, der die Transponder auffordert eine Bitfolge zu senden, die sie eindeutig identifiziert. Dabei handelt es sich meist um die Seriennummer der Transponder, es sind aber auch Anwenderdefinierte Identifikationen möglich. Die Antwort der Transponder erfolgt nach einem Zeitmultiplexverfahren, genauer gesagt dem Slotted-Aloha-Verfahren (siehe oben). Wenn der Reader nun ein kollisionsfreies Signal, also eine eindeutige Identifikations-Bitfolge / Seriennummer empfangen hat, kann er den Transponder mit dieser Seriennummer in den Selected-Modus versetzen und direkt und ausschließlich mit ihm kommunizieren. Nach Abschluss der Kommunikation wird dieser Transponder entweder zurück in den Ready-Modus (in dem sich alle Transponder automatisch befinden nachdem ihre Seriennummer erfolgreich übertragen wurde) oder in den Quiet-Modus versetzt. In diesem Fall wird das Signal dieses

³⁶ [Fink2016], S.262

³⁷ [Kern2011], S.36; [Fink2016], S.259ff; [ISO2009], [FEIG2015], S.41ff

Transponders unterdrückt, um die restlichen Transponder kollisionsfrei erreichen zu können.³⁸

Laut der Bedienungsanleitung des Readers ID ISC.LR1002, der für die Versuche zu dieser Masterarbeit verwendet wurde, kann er dank dieses Antikollisions-Algorithmus bis zu 100 Transponder im Empfangsbereich unterscheiden.³⁹

2.3 Unterscheidungen und Anwendungen

RFID-Systeme lassen sich nach verschiedenen Kriterien unterscheiden und einteilen. Deshalb sind auch deren Anwendungsarten dementsprechend breit gefächert - man findet RFID-Systeme mittlerweile in allen Industriezweigen. Einige der Unterscheidungen und Anwendungen sollen in diesem Kapitel beschrieben werden.

Eine der ältesten und einfachsten Anwendungen ist die Diebstahlsicherung in Kaufhäusern. Dabei kommen die unter Punkt 2.2 beschriebenen 1-Bit Transponder zum Einsatz. Zusätzlich bestehen diese Systeme aus sogenannten Rahmenantennen am Ausgang des Kaufhauses und Deaktivatoren an den Kassen. Solange die meist als Klebeetiketten ausgeführten 1-Bit-Transponder nicht an der Kasse deaktiviert werden, registrieren die mit den großen Rahmenantennen verbundenen Reader, dass der unbezahlte Artikel das Kaufhaus verlässt und es wird Alarm ausgelöst.

Ein weiterer großer Anwendungsbereich sind kontaktlose Chipkarten. Zu deren Anwendung zählt neben dem bargeldlosen Bezahlen beispielsweise der öffentliche Nahverkehr. Hier erfolgt die Einteilung üblicherweise nach der maximal möglichen Entfernung zwischen Reader und Transponder / Chipkarte. Die Einteilung der Chipkartenarten ist Tabelle 3 zu entnehmen.

³⁸[Kern2011], S.36; [Fink2016], S.259ff; [ISO2009]; [FEIG2015], S.41ff

³⁹ [FEIG2015], S.106

	close coupling	proximity coupling	vicinity coupling
Maximaler Leseradius	1 cm	10 cm	1 m
Norm	ISO/IEC 10536	ISO/IEC 14443	ISO/IEC 15693

Tabelle 3 – Chipkartenarten

Eine weitere grundlegende Unterscheidungsmöglichkeit von RFID-Systemen ist die Art der Datenübertragung. Neben den schon oben erwähnten 1-Bit Transpondern, unterscheidet man drei grundsätzliche Prinzipien: das Halbduplexverfahren, das Vollduplexverfahren und das sequentielle Verfahren. Wie aus anderen Bereichen der Datenübertragung bekannt, besteht der Unterschied zwischen Halb- und Vollduplexübertragung darin, dass bei Halbduplex die Datenübertragung in der einen Richtung zeitversetzt zu der anderen Richtung stattfindet. Demnach findet bei der Vollduplexübertragung der sogenannte Uplink und der Downlink parallel statt.⁴⁰ Gemeinsam haben beide Verfahren, dass der RFID-Transponder, vorausgesetzt er befindet sich im Empfangsbereich, stets mit Energie versorgt wird. Dies trifft für RFID-Systeme mit sequentiellen Verfahren nicht zu. Hier wechselt sich die Energieversorgung des Transponders mit der Datenübertragung ab. Demnach müssen diese Transponder Unterbrechungen, beziehungsweise Pausen in der Energieversorgung durch entsprechend dimensionierte Kondensatoren kompensieren können oder es muss sich um aktive Transponder handeln.⁴¹

Es sind noch viele weitere Unterscheidungen von RFID-Systemen möglich. Beispielsweise die verwendete Arbeitsfrequenz, welche unter Punkt 2.1.1 beleuchtet wurde. Das Behandeln aller Merkmale, die man bei der Auswahl eines RFID-

⁴⁰ [Fink2016], S.45f

⁴¹ [Fink2016], S.45f, S.13ff

Systems beachten sollte, würde den Rahmen dieser Arbeit übersteigen. Deshalb werden im Abschnitt 2.3.1 lediglich die Transponderarten genauer betrachtet.

Wie bereits dargestellt, sind die Anwendungsmöglichkeiten von RFID sehr breit gefächert. In Abbildung 7 ist ein Teil der Anwendungsbereiche von RFID, die bisher noch nicht erwähnt wurden, dargestellt.

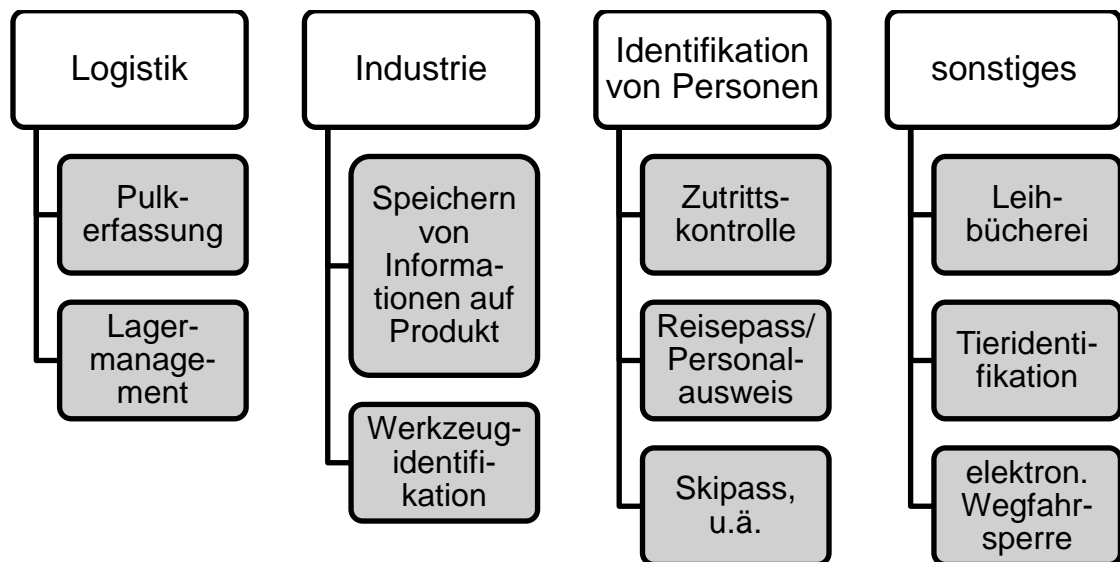


Abbildung 7 – Anwendungsbereiche RFID⁴²

Alle dargestellten Anwendungsbereiche haben die Gemeinsamkeit, dass eine Art von Information auf dem Transponder gespeichert wird und diese vom Lesegerät ausgelesen und - je nach Anwendung - auch schreibend verändert werden. Abgesehen von dieser „klassischen“ Nutzung existieren auch Anwendungen, die über die Identifikation und das Speichern von Informationen hinausgehen. Dazu sind die Stichworte Ortsbestimmung mit RFID und integrierte Messtechnik in RFID-Transpondern zu nennen.

⁴² [Fink2016], S.625ff

2.3.1 Transponderarten

Es gibt unzählige verschiedene RFID-Transponder diverser Hersteller für unterschiedliche Anwendungen auf dem Markt. Unterscheidungskriterien sind beispielsweise: Bauform, Energieversorgung, Speicherkapazität, Programmierfähigkeit (beziehungsweise Beschreibbarkeit).

In der Literatur⁴³ werden zahlreiche Bauformen von Transpondern genannt und erläutert. Im Zuge des Versuchsaufbaus zu dieser Masterarbeit kamen zweierlei Transponderbauformen zum Einsatz: Zum einen Transponder in Disk-, beziehungsweise Münzenbauweise - die am häufigsten verwendete Transponderbauform. Hier befindet sich die Transponderschaltung samt Antenne in einem runden Kunststoffgehäuse. Zum anderen kamen Transponder in Smart-Label-Bauform zum Einsatz. Ein Beispiel für diese Bauform ist in Abbildung 8 zu sehen. Hier werden die Transponderschaltung und die Transponderantenne auf eine dünne Folie aufgebracht. Meistens ist diese Folie selbstklebend – somit kann dieser nur wenige Millimeter dicke Transponder leicht auf jeder beliebigen Oberfläche angebracht werden.

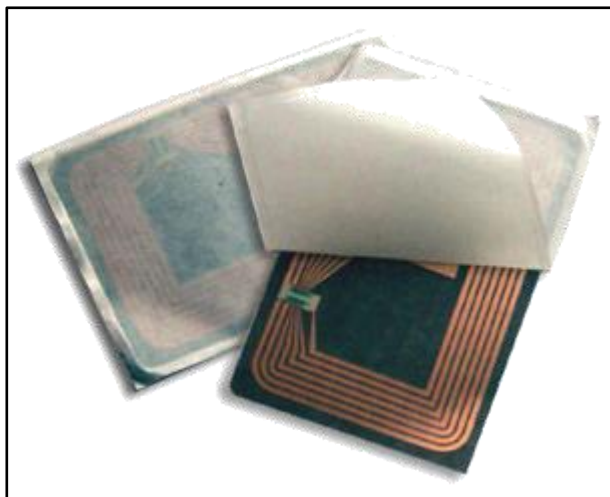


Abbildung 8 – Smart-Label⁴⁴

⁴³ [Fink2016], S.16ff

⁴⁴ [BioE2016]

Die Energieversorgung aller Transponderbauformen bezeichnet man entweder als aktiv oder als passiv. Im Gegensatz zu der, bereits unter Punkt 2.1 beschriebenen, passiven Spannungsversorgung, werden Transponder mit aktiver Spannungsversorgung von einer eigenen Spannungsversorgung gespeist. Diese Aufgabe übernimmt in der Regel eine Batterie. Dadurch wird der Chip in aktiven Transpondern immer mit Spannung versorgt, auch wenn sie sich nicht im Bereich der induktiven Kopplung eines Lesegerätes befinden. Die meisten aktiven Transponder übertragen ihre Informationen jedoch auf die gleiche Weise wie passive Transponder – sie modulieren das Feld des Lesegerätes (siehe Punkt 2.1). Sie produzieren sozusagen kein eigenes modulierte Signal. Aus diesem Grund werden sie als semi-passive Transponder bezeichnet, weil die eigene Spannungsversorgung lediglich zum Betrieb des internen Chips genutzt wird. Die theoretisch mögliche Reichweite dieser Transponder ist höher als bei passiven Transpondern, weil ein weniger starkes Feld für die Kommunikation ausreicht.⁴⁵

2.4 Ortsbestimmung

Die Ortsbestimmung von RFID-Transpondern basiert auf der Messung von Signallaufzeiten. Dazu ist eine geeignete Hardware notwendig. RFID-Systeme, die die Ortsbestimmung ermöglichen, müssen die Signallaufzeit zwischen Transponder und mindestens drei Lesegeräten ermitteln können und diese Informationen an das übergeordnete Rechensystem weitergeben. Abbildung 9 zeigt das Prinzip, wie aus den Signallaufzeiten auf den Standort des Transponders geschlossen werden kann. Dabei wird angenommen, dass sich die Signale zwischen den Transpondern und sämtlichen Readern (in Abbildung 9 als „Receiver“ bezeichnet) mit der gleichen Geschwindigkeit ausbreiten. So kann aus der Signallaufzeit unmittelbar auf die Entfernung zwischen beweglichem Transponder und fest installiertem Reader geschlossen werden. Für den zweidimensionalen Fall (die Z-Koordinate des Transponders ist bekannt und fest) weiß man somit, dass sich der Transponder auf einem gedachten Kreis um den Reader mit dem Radius des ermittelten

⁴⁵ [Fink2016], S.25f

Abstandes befindet. Wenn man nun die Entfernungen des gleichen Transponders zu zwei weiteren Readern ermittelt, kann die Position eindeutig bestimmt werden. Der Transponder befindet sich nämlich im Schnittpunkt der drei gedachten Kreise.

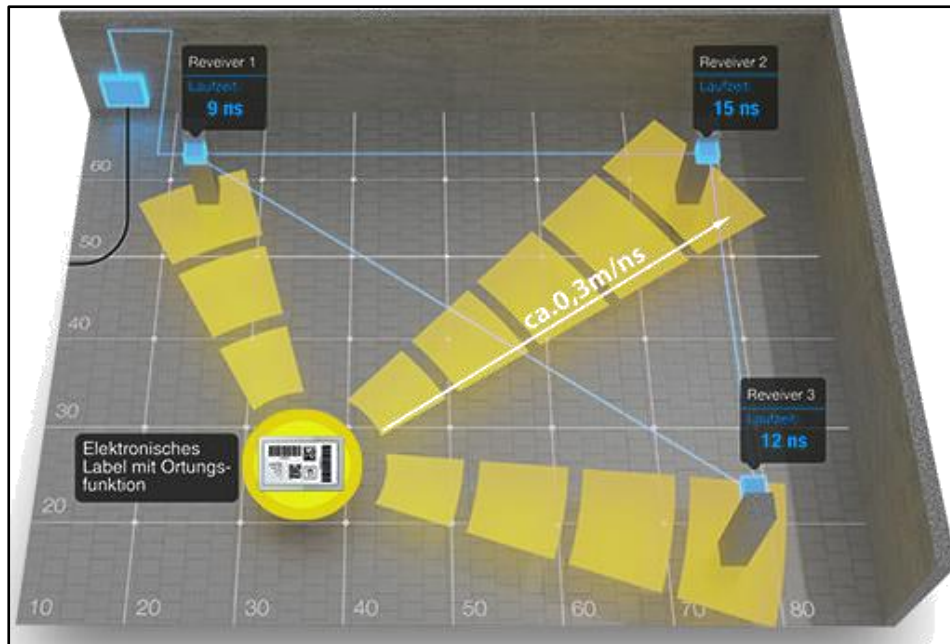


Abbildung 9 – Ortsbestimmung mit RFID⁴⁶

Wenn das RFID-System allerdings keine Information über die Signallaufzeit zwischen Transponder und Reader liefert, ist die Ortsbestimmung komplizierter und weniger exakt. Dies war ebenfalls der Fall beim Versuchsaufbau im Rahmen der Masterarbeit. Deshalb wird darauf in einem der nachfolgenden Abschnitte eingegangen.

⁴⁶ [Schn16]

3 Versuchsaufbau

Wie bereits erwähnt, wurde im Rahmen dieser Arbeit ein Industrieroboter mit vergleichsweise einfacher RFID-Technik kombiniert. In diesem Abschnitt soll genauer darauf eingegangen werden, wie diese beiden Teilsysteme erfolgreich miteinander verknüpft werden können. Zuvor werden jedoch die Aufbauten beider Systeme einzeln betrachtet.

3.1 RFID-System

Der RFID-Versuchsaufbau besteht aus einem Lesegerät, einer Antenne und zweierlei Arten von Transpondern, von denen jeweils mehrere zur Verfügung stehen. Diese Bestandteile sollen in diesem Abschnitt nacheinander behandelt werden.

3.1.1 Schreib-/ Lesegerät

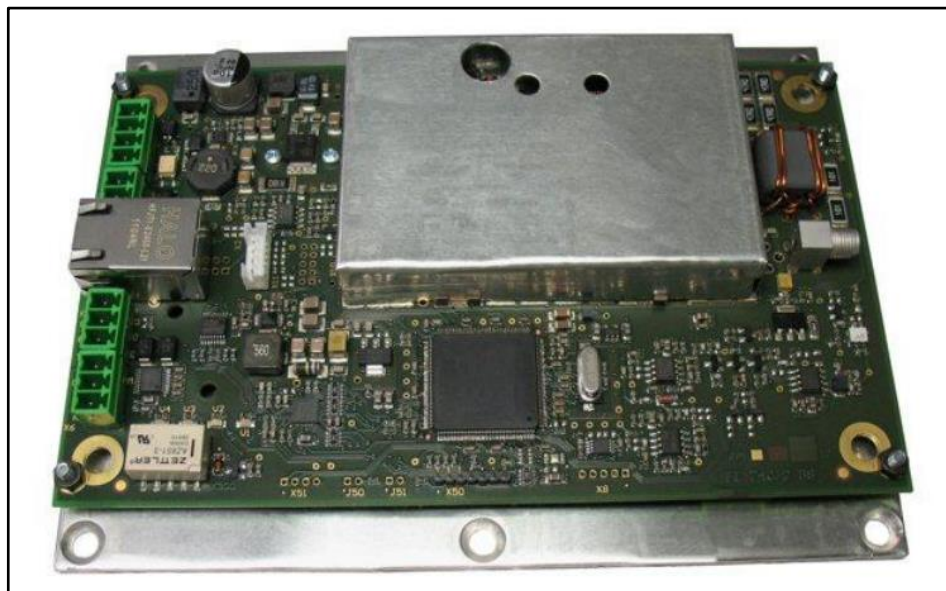


Abbildung 10 – Reader⁴⁷

⁴⁷ [FEIG2014], S.1

Abbildung 10 zeigt das Schreib-/ Lesegerät, das für den Versuchsaufbau im Rahmen dieser Masterarbeit verwendet wurde. Es handelt sich um das Modell ID ISC.LRM1002, ein Long Range Reader Modul der Feig Electronic GmbH. Dieser Reader verfügt über die Schnittstellen RS232, USB und Ethernet (TCP/IP), über welche er mit einem Computer kommunizieren kann. In Abschnitt 3 der Bedienungsanleitung⁴⁸ des Readers ist aufgeführt, welche Schnittstelle am besten für verschiedene Anwendungsarten geeignet ist. In dieser Masterarbeit wurde die Kommunikation ausschließlich über Ethernet geführt, weil sich der Reader so perfekt in das Gesamtsystem – siehe Abbildung 18 – integrieren ließ. Ferner ist die Kommunikationszeit zwischen Reader und Computer über Ethernet ausreichend für die Anwendungen, die für diese Masterarbeit relevant sind. Die Kommunikationszeit wird – neben der Wahl der einzelnen Komponenten – unter anderem durch die Transponderanzahl im Empfangsbereich bestimmt.⁴⁹ Um Problemen durch unterschiedliche Antwortzeiten im späteren Programmablauf vorzubeugen, wurde geprüft, wie stark die Transponderanzahl die Kommunikationszeit beeinflusst. Dazu wurde pro Transponderanzahl 100-mal die Zeit zwischen Befehl (des Computers) und Antwort (des Readers) mit Hilfe von Matlab gemessen. Das Ergebnis ist im Diagramm in Abbildung 11 dargestellt. Es ist zu erkennen, dass die Kommunikationszeit relativ konstant bleibt, solange sich weniger als neun Transponder im Empfangsbereich befinden. Sobald die Transponderanzahl größer als acht wird, steigt die Kommunikationszeit. Die dargestellte Kurve verläuft weder linear, noch exponentiell. Man kann allerdings schlussfolgern, dass mit steigender Transponderanzahl auch die Kommunikationszeit steigt, jedoch weder proportional, noch nach einer anderen mathematischen Beziehung. Der Grund für den Zusammenhang von Kommunikationszeit und Transponderanzahl ist das gesteigerte Auftreten von Kollisionen – siehe Abschnitt 2.2.4 – bei höherer Transponderanzahl.⁵⁰

⁴⁸ [FEIG2015]

⁴⁹ [FEIG2015], S.165

⁵⁰ [FEIG2015], S.165

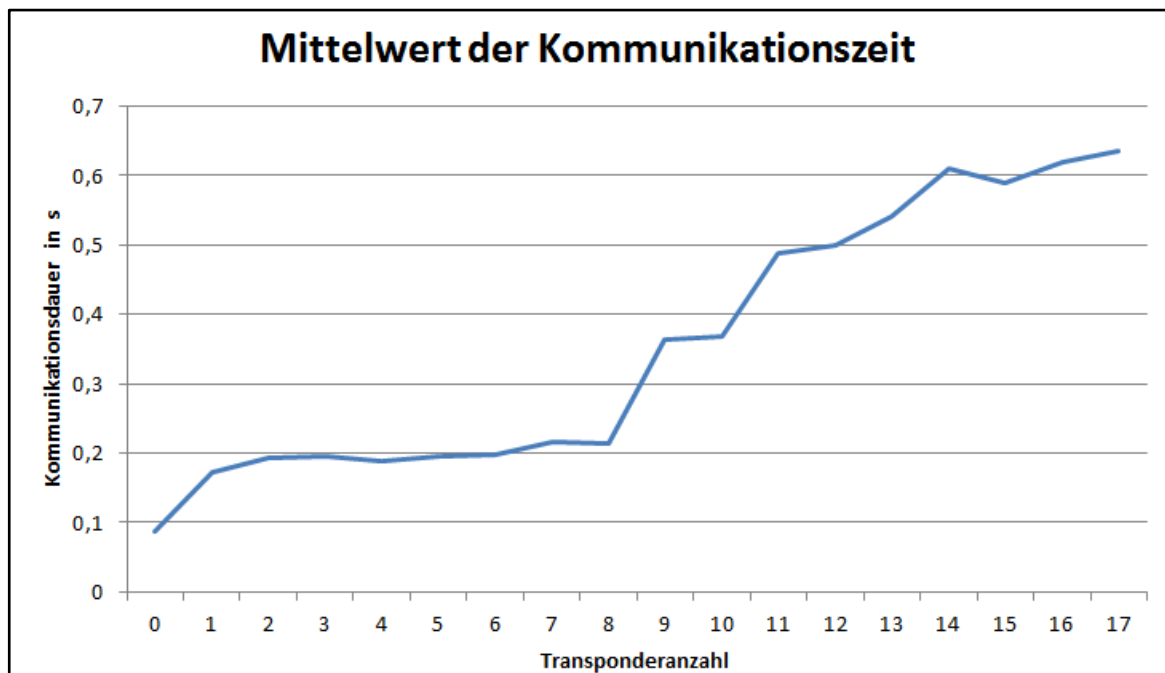


Abbildung 11 – Kommunikationszeit in Abhängigkeit der Transponderanzahl

Der Reader arbeitet auf einer Frequenz von 13,56 MHz. Damit ist er geeignet für den Betrieb mit Transpondern nach ISO 15693. Um das Schreib-/ Lesegerät in Betrieb zu nehmen, muss eine vom Hersteller unterstützte Antenne am Antennenanschluss des Readers (Rechts in Abbildung 10) angeschlossen werden. Die Ausgangsleistung, die in die Antenne eingespeist wird kann zwischen 1 W und 5 W in Schritten von 1 W eingestellt werden. Diese und unzählige andere Einstellungen und Funktionen können über die mitgelieferte Software parametrieren werden. Da diese Software (bei Wahl von Ethernet als Kommunikationsschnittstelle) die Anweisungen in Folgen von Hexadezimalzahlen codiert und diese Folgen über die gewählte Schnittstelle zum Reader sendet, kann diese Kommunikation ebenso von einer anderen Software wie Matlab ausgehen. Mehr dazu ist in Abschnitt 4.1 zu lesen.

3.1.2 Antennen

Zum ersten Versuchsaufbau dieser Masterarbeit gehört die Hochfrequenzantenne ID ISC.ANT310310-A der Firma RFID Canada. Die Antenne zeichnet sich durch eine große Reichweite aus. Dies geht mit vergleichsweise großen Abmessungen

einher. In Abbildung 12 ist die Antennenreichweite in Abhängigkeit von der Sendeleistung aufgeführt. Dabei ist zu beachten, dass diese Daten unter idealen Bedingungen und mit zwei größeren Transpondern (46 mm x 75 mm) als jenen, die im Rahmen der Masterarbeit verwendet wurden, aufgenommen wurden. Diese Werte konnten mit dem verwendeten RFID-System demzufolge nicht erreicht werden, auch weil der Reader, wie oben beschrieben, eine maximale Sendeleistung von 5 W hat. Außerdem haben einige weitere Faktoren Einfluss auf die Reichweite. Wie in der Montageanleitung der Antenne nachzulesen ist, wird die Reichweite neben der Sendeleistung auch von der Anwesenheit von Metall in der näheren Umgebung beeinflusst. Demnach sollte beachtet werden, dass der Mindestabstand von 5 cm zwischen Antenne und Metallteilen nicht unterschritten wird. Außerdem ist in Entfernungen bis 30 cm mit starken Verschlechterungen der Reichweite zu rechnen. Erst ab einer Entfernung von 50 cm ist der Einfluss von Metallteilen vernachlässigbar gering. Ist der Einsatz von Metall in der nahen Umgebung dennoch nicht umgänglich, ist es durch einen Abgleich der Antenne möglich, den Auswirkungen zu reduzieren.⁵¹

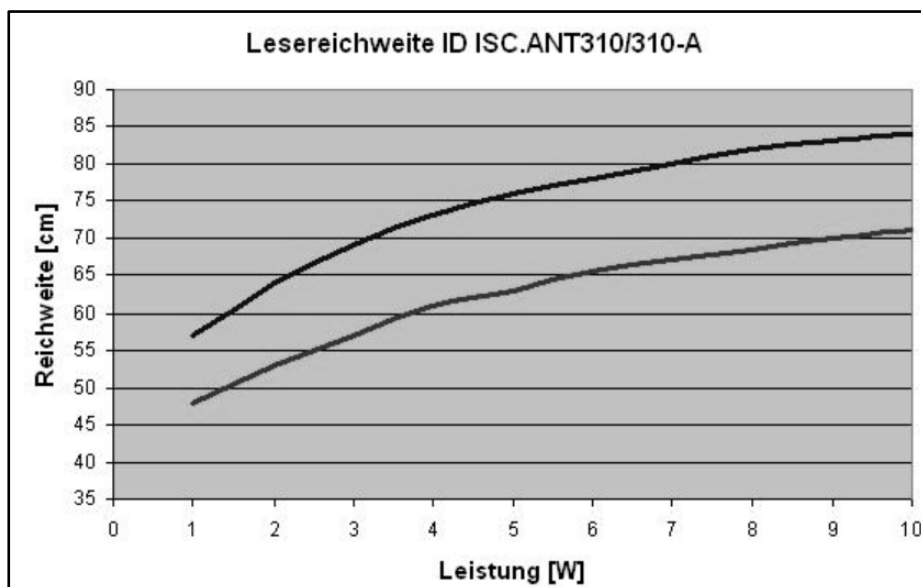


Abbildung 12 – Lesereichweite zweier Transponder in Abh. der Sendeleistung⁵²

⁵¹ [FEIG2009], S.13

⁵² [FEIG2009], S.12

Neben der Entfernung zur Antenne, ist die richtige Ausrichtung beziehungsweise Orientierung des Transponders gegenüber der Antenne eine Voraussetzung für die störungsfreie Kommunikation. Wenn sich ein Transponder beispielsweise genau in der Antennenmitte befindet – dort ist die Dichte der Feldlinien am höchsten – ist dies also keine Garantie dafür, dass der Transponder erkannt wird. Der Grund dafür ist, dass der Transponder bei senkrechter Orientierung (beziehungsweise näherungsweise senkrechter Orientierung) nicht mit ausreichend vielen Feldlinien durchsetzt wird und somit keine Kommunikation und Energieversorgung möglich ist. In Abbildung 13 ist der Feldlinienverlauf der verwendeten Antenne dargestellt. Dort ist zu sehen, dass die Feldliniendichte und die Reichweite der Antenne über (und unter) der Mitte der Antenne am höchsten ist. In den Orientierungen zwischen den beiden Extremen nehmen die Reichweite und die Dichte der Feldlinien mit spitzer werdenden Winkel zwischen Transponder und Antennenfläche zu.⁵³

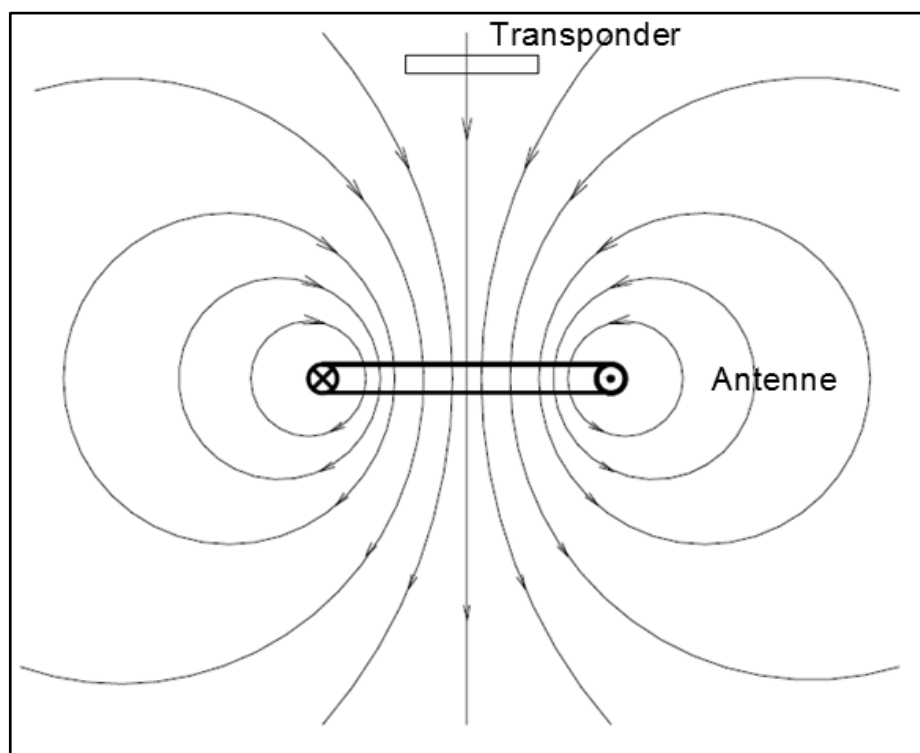


Abbildung 13 – magnetische Feldlinien der Antenne⁵⁴

⁵³ [FEIG2009], S.20

⁵⁴ [FEIG2009], S.20

Im Rahmen der Versuche zu dieser Masterarbeit wurden in zahlreichen Messungen Diagramme erstellt, die aufzeigen, an welchen Transponderpositionen eine Kommunikation zu Stande kommen kann. Diese Diagramme haben die Grundlage für die Transponderortung gebildet, weshalb unter Punkt 4.2 genauer darauf eingegangen wird.

Neben der angesprochenen Transponderortung wurden im Rahmen dieser Masterarbeit weitere Aufgaben, wie das intelligente Greifen (Abschnitt 4.3), realisiert. Dafür ist die große Rahmenantenne weniger geeignet. Deshalb kann, je nach Aufgabe, die Antenne durch kleinere Exemplare ersetzt werden. Zum Einsatz kommt die ID ISC.ANT40/30 (rechts in Abbildung 14) oder ID ISC.ANT100/100 (links in Abbildung 14). Diese kleineren Antennen sind sogenannte Platinen-Antennen des Herstellers RFID Canada. Die Leiterschleife und alle anderen Bauteile der Antennen sind, wie in Abbildung 14 zu sehen, in eine Platine eingelassen. Die Zahlenangaben in den Antennenbezeichnungen stehen für die Abmessungen der Platine in Millimeter.

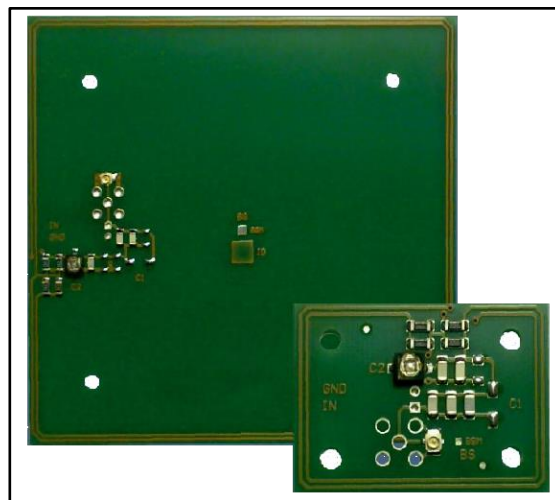


Abbildung 14 – Platinen-Antennen⁵⁵

Diese beiden Antennen dürfen jeweils mit maximal einem Watt Sendeleistung betrieben werden. Deshalb und wegen der viel kleineren Abmessungen der Leiterschleifen im Vergleich zur Antenne ID ISC.ANT310310-A, ist die Reichweite der

⁵⁵ [RFID2014], S.2

Platinen-Antennen um einiges geringer als die Reichweite der Rahmenantenne. So hat die kleinere Antenne (ID ISC.ANT40/30) laut Datenblatt⁵⁶ eine Reichweite von 7 cm und die größere Antenne (ID ISC.ANT100/100) eine Reichweite von 14 cm. Diese Reichweiten sind für das intelligente Greifen vollkommen ausreichend und werden durch die gleichen Faktoren beeinflusst wie die Reichweite der Rahmenantenne.

3.1.3 Transponder

Zusammen mit den bereits erläuterten Bestandteilen des RFID-Systems wurden für den Versuchsaufbau zu dieser Masterarbeit zweierlei Transponderarten geordert. Beide wurden von der Firma Hans Turck GmbH & Co.KG gefertigt. Außerdem haben die Transponder selbstverständlich gemeinsam, dass sie die gleiche Arbeitsfrequenz wie der Reader – 13,56 MHz – haben, die Datenübertragung auf induktiver Kopplung basiert und sie nach ISO 15693 standardisiert sind. Diese drei Faktoren bilden die Voraussetzung dafür, dass eine Kommunikation mit dem gewählten, unter Punkt 3.1.1 erläuterten, Schreib-/ Lesegerät möglich ist.⁵⁷

Der erste Transpondertyp ist der TW-R20-K2 (links in Abbildung 15), ein Transponder in Disk-Bauweise (siehe Abschnitt 2.3.1). Seine Bezeichnung setzt sich wie folgt zusammen⁵⁸:

- Das „T“ steht für den Hersteller „TURCK RFID-System BL ident®“
- Das „W“ steht für „read/write“, also für lesenden und schreibenden Speicherzugriff.
- „R20“ steht für die Bauart und Baugröße: rund, 20mm Durchmesser.
- „K2“ steht für die Speichergröße von 2 Kilobyte.

Bei dem eingebautem Chip handelt es sich um einen Fujitsu MB89R118 und der Speicher des Transponders ist als FRAM ausgeführt. Von dessen 2 Kilobytes

⁵⁶ [RFID2014]

⁵⁷ [Tur2016a], [Tur2016c], S.11

⁵⁸ [Tur2016c], S.11

Speicherkapazität, also 2048 Bytes sind 2000 Bytes frei nutzbar, also vom Benutzer beschreibbar. Die 2000 Bytes sind in 250 Blöcke mit je 8 Bytes gegliedert und auf diese Weise werden sie auch adressiert.⁵⁹



Abbildung 15 – Transponder

Im Gegensatz zu dem TW-R20-K2 handelt es sich bei dem Transponder mit der Bezeichnung TW-L50-50-P-B128 (rechts in Abbildung 15) um einen Smart-Label-Transponder (siehe Abschnitt 2.3.1). Die ersten beiden Buchstaben der Bezeichnung haben diese Transponderarten gemeinsam. Die übrigen Zeichen der Bezeichnung setzen sich wie folgt zusammen⁶⁰:

- „L50-50“ steht für die Bauart und Baugröße: Label, 50 mm x 50 mm.
- „P“ steht für das Merkmal „paper data carrier“ – Papierdatenträger
- „B128“ steht für die Speichergröße von 128 Byte.

Auch was den eingebauten Chip und den Speicher angeht, unterscheidet sich der Smart-Label-Transponder vom Disk-Transponder. So kommt im TW-L50-50-P-B128 ein Chip von NXP-Semiconductors zum Einsatz. Der 128 Byte große Speicher ist als EEPROM realisiert. Analog zum TW-R20-K2 steht nicht der gesamte Speicher zur freien Verfügung für den Anwender. 112 Byte sind frei nutzbar. Diese

⁵⁹ [Tur2016a]

⁶⁰ [Tur2016c], S.11

sind in 28 Blöcke à 4 Bytes gegliedert und werden ebenso adressiert. Damit unterscheidet sich die Adressierung bei beiden Transponderarten. Dies muss bei der Programmierung berücksichtigt werden. Wie genau die Daten der Transponder adressiert, ausgelesen und beschrieben werden, wird in Kapitel 4.1.2 erläutert.⁶¹

3.2 Industrieroboter

Das zweite Teilsystem des Versuchsaufbaus dieser Masterarbeit bildet ein Industrieroboter der Firma ABB und ein am Roboter montierter, taktiler Greifer – ein WSG 50 der Firma Weiss Robotics. Der Roboter ist in Abbildung 16 zu sehen.

Der Roboter IRB 120 kam zum Einsatz, weil er bereits für andere Projekte der Hochschule verwendet wird und sich somit bereits bewährt hat. Der Hersteller beschreibt sein Produkt folgendermaßen: „Der IRB 120 ist ABB’s kleinster Industrieroboter. Er ist vielseitig einsetzbar, wiegt nur 25 kg, hat eine Handhabungskapazität von 3 kg (bis zu 4 kg mit vertikalem Handgelenk) und eine Reichweite von 580 mm. Der Roboter ist eine kosteneffiziente Lösung für hohe Produktionsleistung. Er eignet sich hervorragend für eine flexible Produktion auf engstem Raum.“⁶²

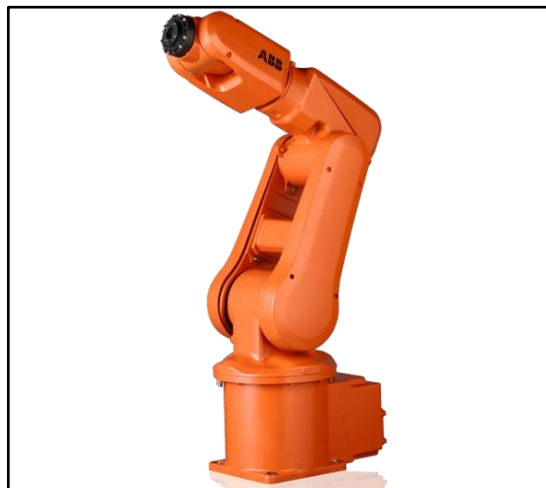


Abbildung 16 – IRB 120⁶³

⁶¹ [Tur2016b]

⁶² [ABB2016]

⁶³ [ABB2016]

Wie bereits erwähnt, wurde der Roboter bereits für andere Projekte an der Hochschule im Vorfeld dieser Masterarbeit verwendet. Er wurde also schon in Betrieb genommen und konnte direkt verwendet werden. Aus diesem Grund soll an dieser Stelle für weitere Informationen zum verwendeten Roboter auf dessen Bedienungsanleitung⁶⁴ verwiesen werden. Der Roboter kommuniziert über Ethernet und kann so gesteuert werden.

3.2.1 Greifer

Wie bereits erwähnt, wurde der Roboter mit einem Greifer der Firma Weiss Robotics ausgestattet. Der Greifer WSG 50⁶⁵, welcher bereits, wie der Roboter, vor Beginn dieser Masterarbeit für andere Projekte genutzt wurde, ist ein servoelektrischer Parallelgreifer. Wie in Abbildung 17 zu sehen ist, können sich die parallel gegenüberliegenden Finger auf einer Geraden aufeinander zu, sowie voneinander wegbewegen und somit das Objekt greifen beziehungsweise loslassen. Die beweglichen, schwarzen Teile in Abbildung 17 sind allerdings nicht die Finger. An diese hier sichtbaren Teile können verschiedene Finger montiert werden. Somit ist der Greifer dann universell einsetzbar. In der Ausführung, die im Rahmen dieser Arbeit eingesetzt wurde, wird der Greifer durch zwei Finger mit taktiler Sensorik ausgestattet, sie können also die Oberflächen der gegriffenen Objekte ertasten. Diese Funktion blieb allerdings ungenutzt, deshalb soll an dieser Stelle nicht näher darauf eingegangen werden. Unabhängig von der Wahl des Fingertyps, hat der Greifer einen Hub von 110 mm. Das bedeutet, dass er Objekte bis zu einer maximalen Abmessung von 110 mm greifen kann. Dabei sind die Greifkraft, die Greifgeschwindigkeit und die Beschleunigungen der Finger vom Anwender stufenlos wählbar. Die Wahl dieser Parameter ist lediglich durch Grenzwerte limitiert. So ist die Greifkraft zwischen 5 N und 80 N einstellbar, die maximale Geschwindigkeit beträgt 420 mm/s und die Beschleunigung hat ihr Maximum bei 5000 mm/s². Die-

⁶⁴ [ABB2013]

⁶⁵ [Weis2014]

se und weitere technische Daten sind der Montage- und Bedienungsanleitung⁶⁶ des Greifers zu entnehmen.

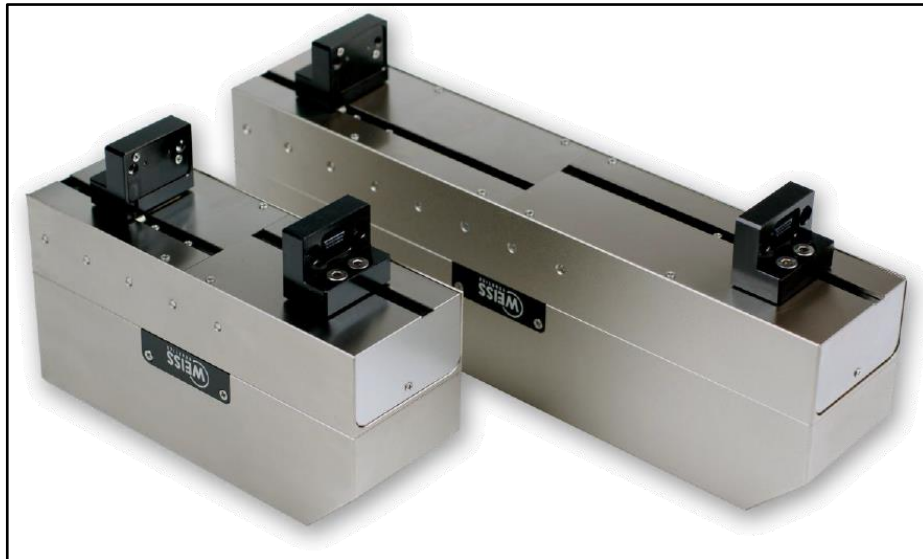


Abbildung 17 – Greifer WSG 50⁶⁷

3.3 Gesamtsystem

Abbildung 18 stellt die Beziehungen der einzelnen Teilsysteme dar. Jedes dieser Teilsysteme ist in einer eigenen Farbe dargestellt. Außerdem sind die Kommunikationswege zwischen allen Komponenten abgebildet. Die in den Kapiteln 3.1 und 3.2 erläuterten Komponenten haben alle ihre eigene, vom jeweiligen Hersteller mitgelieferte Software zur Parametrierung und Steuerung. Sämtliche Funktionen des RFID-Systems und des ABB-Roboters können über eine eigene mitgelieferte Software in Betrieb genommen und gesteuert werden. Der Greifer lässt sich dagegen über ein Web-Interface ansteuern. Gemeinsam haben alle Teilsysteme, dass die Kommunikation zwischen Steuerungscomputer und jeweiligem Gerät über Ethernet stattfindet.

⁶⁶ [Weis2014]

⁶⁷ [Weis2014], S.1

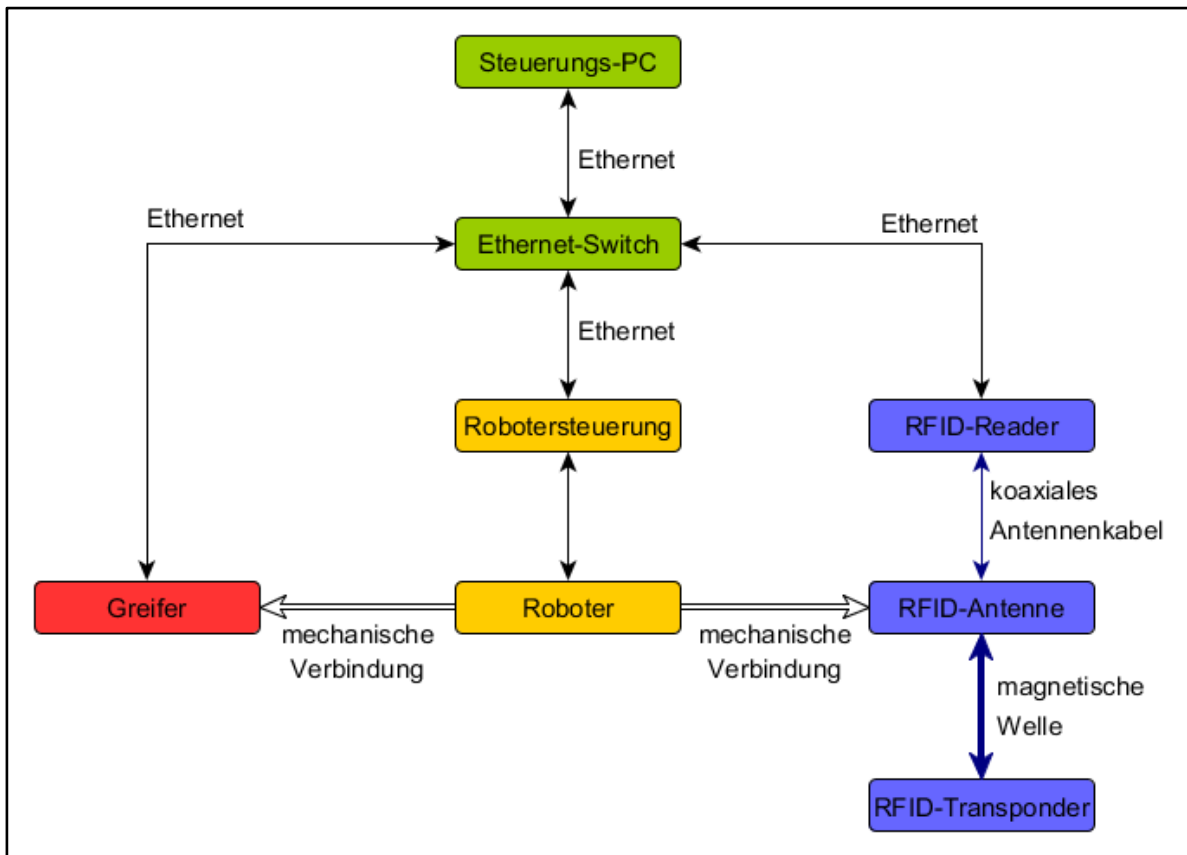


Abbildung 18 – Gesamtsystem

4 Durchführung

Hier wird erläutert, wie die gestellten Aufgaben analysiert, bearbeitet und gelöst wurden. Außerdem wird auf aufgetretene Probleme hingewiesen und es werden Lösungsvorschläge präsentiert.

4.1 Matlab

Matlab ist ein Simulationstool des amerikanischen Unternehmens MathWorks. MathWorks beschreibt sein Produkt folgendermaßen: „MATLAB® ist eine hochentwickelte Sprache und interaktive Umgebung, die von Millionen Entwicklern und Wissenschaftlern weltweit genutzt wird. In dieser Umgebung können Sie Ihre Ideen entwickeln und darstellen und außerdem disziplinübergreifend arbeiten. Das schließt Signal- und Bildverarbeitung, Kommunikation, Steuerungs- und Regelsysteme sowie finanzmathematische Operationen mit ein.“⁶⁸ Die Matlab-Umgebung⁶⁹ hat eine eigene Programmiersprache, auf deren Basis sie ein Programmpaket für eine Vielzahl von Anwendungen – nicht nur zur numerischen Berechnung und Visualisierung – bietet.

Für die Versuche während dieser Masterarbeit wurde MATLAB verwendet, um, wie oben bereits erwähnt, die Funktionen des RFID-Systems zu steuern. Dies wäre allerdings auch mit der zum RFID-Reader mitgelieferten Software möglich gewesen. Der Hauptgrund für den Einsatz von Matlab war also ein anderer: Mit Matlab wurde es möglich die Steuerung des Industrieroboters und des RFID-Systems in einer Programmierumgebung zusammenzufassen. Dadurch wurde es möglich, Anweisungen an beide Teilsysteme aus ein und demselben Programm heraus zu senden. Es konnten also auch teilsystemübergreifende Abhängigkeiten, wie bei-

⁶⁸ [Math2016]

⁶⁹ [Piet2014], S.1f

spielsweise das Auslesen eines Transponders – sobald der Roboter eine bestimmte Position erreicht hat – programmiert werden.

4.1.1 Struktur der Datenpakete des RFID-Systems

Die Übernahme der Funktionen des RFID-Readers in die Matlab-Programmierungsumgebung hat einen großen Teil der Arbeit am Versuchsaufbau zu dieser Masterarbeit ausgemacht. Zuvor musste jedoch geklärt werden, wie die Informationen und Kommandos codiert sind. Generell gilt, dass der Reader nicht selbstständig sendet, er antwortet lediglich auf Anfragen des Host-Computers.⁷⁰

Byte	Host → Reader	Reader → Host
1	STX – Startbyte, immer 0x02	
2	MSB ALENGTH – Position Startbyte STX, immer 0x00	
3	LSB ALENGTH – Byteanzahl des gesamten zu übertragenden Pakets	
4	COM-ADR – Zieladresse des Pakets (0x00 \triangleq Computer, 0xFF \triangleq Reader)	
5	Control-Byte – beinhaltet Befehl, den Reader ausführen soll	
6	Request-Data – beinhaltet eigentliche Information zur Anfrage, z.B. Seriennummer Transponder;	Statusbyte – Status des Readers
7 ... n-2	variable Länge	Response-Data – beinhaltet eigentliche Information / Antwort auf Anfrage; variable Länge
n-1 n	CRC-16 Prüfsumme – gebildet aus den Bytes von 1 bis n-2 mit dem CRC-Polynom $x^{16}+x^{12}+x^5+1$	

Tabelle 4 – Struktur der Datenpakete⁷¹

⁷⁰ [Feig2015], S.102

⁷¹ [Feig2015], S.102

Wie bereits erwähnt, kommunizieren Computer und Reader mittels Folgen von Hexadezimalzahlen, die über die gewählte Schnittstelle übertragen werden. Diese Folgen haben stets die gleiche Struktur. Aus diesem Grund liefert Tabelle 4 eine Übersicht über den Aufbau der Datenpakete.⁷²

Sämtliche Datenpakete beginnen mit dem sogenannten Startbyte – einer „2“ in Hexadezimalschreibweise. Das signalisiert dem jeweiligen Empfänger, dass eine Information beziehungsweise ein Kommando folgt. Neben diesem Byte sind 6 weitere Bytes in jeder hexadezimalen Ziffernfolge enthalten. Byte 2 und 3 geben Information über die Länge des zu empfangenden Paketes. Byte 4 beinhaltet die Adresse des Teilnehmers für den das Datenpaket bestimmt ist. Für das in dieser Arbeit beschriebene RFID-System gab es für dieses Byte lediglich zweierlei Werte: 0x00 – für den Computer (Host) und 0xFF (hexadezimale 255) – für den Reader. Das 5. Byte ist spezifisch für jeden Befehl, den der Computer an den Reader sendet oder auf den der Reader antwortet. Dabei ist dieses Byte in der Anfrage und deren Antwort stets identisch. Dank dieses Bytes „wissen“ Reader und Host-Computer wie die nachfolgenden Daten interpretiert werden müssen. Die letzten beiden Bytes, die jedes Datenpaket enthält sind für die Sicherung der Kommunikation zuständig. Das bedeutet, dass anhand dieser zwei Bytes erkannt werden kann ob die Übertragung des Datenpakets fehlerfrei war. Dazu beinhalten diese beiden Bytes eine CRC-Prüfsumme aus allen Bytes des Datenpakets (abgesehen von den letzten beiden Bytes), welche im sendenden Teilnehmer berechnet wurde und mit übertragen wird. Das CRC-Verfahren ist in der Literatur⁷³ detailliert und verständlich erklärt. Deshalb werden in dieser Arbeit nur die wichtigsten Eckpunkte genannt, es wird hier nicht detailliert darauf eingegangen werden. CRC steht für cyclic redundancy check –zyklische Redundanzprüfung. Die Prüfsumme wird im sendenden Teilnehmer mit dem CRC-Prüfpolynom $x^{16}+x^{12}+x^5+1$ berechnet und übertragen. Nur wenn im Empfänger die Polynomdivision aus übertragenem Da-

⁷² [Feig2015], S.102

⁷³ [GeKr2015]

tenpaket (inklusive CRC-Prüfsummenbytes) und dem Polynom ohne Rest aufgeht, war die Übertragung erfolgreich und nur dann wird das Datenpaket akzeptiert.⁷⁴

Wie in Tabelle 4 zu erkennen ist, wird ein Statusbyte nur in den Antworten des Readers übertragen. Die Kommunikation wird nur dann fortgesetzt, wenn der Reader über das Statusbyte meldet, dass alles in Ordnung ist.⁷⁵

4.1.2 Implementierte Funktionen des RFID-Systems

Nachdem sich mit der Struktur der Datenpakete vertraut gemacht wurde, konnten nach und nach sämtliche Funktionen des RFID-Systems, die benötigt wurden, in Matlab programmiert werden. Zuerst wurden die Grundfunktionen übernommen. Diese mussten sozusagen lediglich „nachprogrammiert“ werden. Dazu zählte, neben dem Verbindungsaufbau und dem Beenden der Kommunikation, auch ein Befehl, der die Transponderanzahl im Empfangsbereich liefert, sowie ein Befehl, dessen Antwort Informationen über den angeschlossenen Reader liefert. Diese Befehle haben gemeinsam, dass für deren Umsetzung lediglich statische Hexadezimalzahlen über Ethernet an den Reader gesendet werden müssen. Dieser antwortet dann auf gleiche Weise. Die gesendete Anfrage ist also bei jedem dieser Befehle stets jeweils die gleiche Zahlenfolge. Ebenso hat die Antwortfolge stets mindestens die gleiche Struktur und Länge. Zur Verdeutlichung ist hier der Algorithmus, der die Transponderanzahl im Empfangsbereich liefert, dargestellt:

⁷⁴ [FEIG2015], S.24; [GeKr2015], S.23ff

⁷⁵ [FEIG2015], S.102

```
function [anzahl] = RFID_anzahl(rfid)
%Gibt die Transponderanzahl im Empfangsbereich zurück
%RFID-Handle-Variable, welche bei RFID_start erstellt wird, muss
%übergeben werden

rfid=sende_2(rfid,[2,0,9,255,176,1,0,24,67]);
[rfid,x]=empfang_2(rfid,8);
disp('Anzahl Transponder im Empfangsbereich');
if x(6)~=0
    disp('kein Transponder im Empfangsbereich');
    anzahl=0;
else
    anzahl=x(7);
    [rfid,x]=empfang_2(rfid,((9+(anzahl*10))-8));
    disp(num2str(anzahl));
end
end
```

In diesem kurzen Programm wird zuerst mit Hilfe der selbstgeschriebenen Funktion „sende_2“ die Zeichenfolge „2, 0, 9, 255, 176, 1, 0, 24, 67“ übertragen (siehe Tabelle 4 für die Bedeutungen einzelnen Stellen). Diese wird von Matlab automatisch in Hexadezimalzahlen umgewandelt und fordert den Reader auf, den Inventur-Befehl auszuführen. Im nächsten Schritt werden – durch Aufruf einer weiteren selbsterstellten Funktion – die ersten 8 Zeichen der Antwort des Readers empfangen. Wie bereits aus Tabelle 4 bekannt, befindet sich in Byte 6 der Status des Readers. Deshalb wird anhand dieses Bytes der weitere Programmablauf differenziert. Wenn sich kein Transponder im Empfangsbereich befindet, übermittelt der Reader einen Fehler, sein Status (Byte 6) ist ungleich 0. In diesem Fall wird die Variable „anzahl“ auf 0 gesetzt, es wird im command window „kein Transponder im Empfangsbereich“ ausgegeben und die Funktion wird ohne Ausführung der Befehle unter „else“ beendet. Sollte Byte 6 allerdings den Wert 0 annehmen, dann befindet sich mindestens ein Transponder im Empfangsbereich und somit werden die Befehle unter „else“ ausgeführt. Byte 7 enthält in diesen Fällen die Anzahl der Transponder im Empfangsbereich. Diese wird in die Variable „anzahl“ geschrieben und im command window ausgegeben. Damit weitere Befehle über Ethernet an den Reader gesendet werden können und deren Antworten richtig interpretiert werden können, müssen die restlichen Bytes der Antwort des Readers ebenfalls ausgelesen werden – auch wenn sie gar nicht genutzt werden. Die Länge dieses Antwortdatenpakets hängt von der Transponderanzahl ab. Deshalb musste die, an die Funktion „empfang_2“ übergebene Byteanzahl, variabel ausgeführt sein, was mit „((9+(anzahl*10))-8)“ umgesetzt wurde.

Komplexer wird die Programmierung der RFID-Befehle, wenn sich der Befehl, der von Matlab ausgeht, ebenfalls von Aufruf zu Aufruf unterscheiden kann. Verantwortlich für diese Veränderungen sind beispielsweise die Transponderanzahl im Empfangsbereich und die verschiedenen Seriennummern der Transponder – über diese werden die Transponder nämlich adressiert. Hinzu kommt, dass durch Änderung dieser Faktoren, also der Datenbytes (siehe Tabelle 4) ebenfalls die CRC-Bytes am Ende der Datenpakete variieren. Aus diesem Grund musste auch die CRC-Berechnung als Funktion in die Matlab-Programmierungsumgebung aufgenommen werden. Die Programmierung dieser Funktion basiert auf dem in der Systemanleitung des Readers⁷⁶ beschriebenen und aufgeführten CRC-Algorithmus und ist in den folgenden Zeilen aufgeführt:

```
function [ crc16 ] = crc(daten)
%Berechnet die CRC-Prüfbytes
%Die zu sendenden Daten (Bytes 1 bis n-2) müssen der Funktion
%übergeben werden

crc16=65535;
%diese Zahl entspricht dem Startwert in dezimaler Schreibweise
for i=1:length(daten)
    crc16=bin_dec(xor(dec_bin(crc16),dec_bin(daten(i))));
    for j=1:8
        temp=dec_bin(crc16);
        if (temp(16)==1)
            crc16=bin_dec(xor(dec_bin(fix(crc16/2)),dec_bin(33800)));
            %33800 entspricht dem Prüfpolynom in dez. Schreibweise
        else
            crc16=fix(crc16/2);
        end
    end
end
end
end
```

Sämtliche Funktionen in dieser Funktion, außer *dec_bin* und *bin_dec*, sind standardmäßig in Matlab integriert. Die beiden benutzererstellten Funktionen wandeln Zahlen aus dem dezimalen System in das binäre System und umgekehrt. Da dieser Algorithmus bis auf diese Ausnahmen nicht selbst erstellt wurde, sondern lediglich aus der Programmiersprache C, in welcher er in der Systemanleitung des

⁷⁶ [FEIG2015], S.23f

Readers⁷⁷ aufgeführt ist, übersetzt wurde, soll an dieser Stelle nicht näher darauf eingegangen werden.

Beide CRC-Funktionen (die in der Anleitung dargestellte und die in Matlab übersetzte) haben gemeinsam, dass sie eine Zahl (im Folgenden „CRC-Zahl“ genannt) ausgeben. Aus dieser Zahl können anschließend einfach die beiden CRC-Prüfbytes berechnet werden. Dazu wird für das erste Prüfbyte (Byte n-1 in Tabelle 4) die Modulo-Funktion mit dem Divisor 256 auf die ausgegebene CRC-Zahl angewandt. Das Ergebnis ist der Rest aus der Division der CRC-Zahl mit 256. Das zweite Prüfbyte (Byte n in Tabelle 4) wird mit dem Quotienten der ganzzahligen Division der CRC-Zahl mit 256 beschrieben. In der Matlab-Programmiersprache sehen diese Operationen folgendermaßen aus: Byte (n-1): „`mod(crc16,256)`“, Byte n: „`fix(crc16/256)`“.

Somit war auch die Übernahme der typischen RFID-Funktionen möglich. Aus dem Matlab-Kommandofenster heraus kann man sich die Anzahl der Transponder im Empfangsbereich sowie deren Seriennummern ausgeben lassen. Ebenso ist es möglich, die Transponder einzeln – anhand der Seriennummer – zu adressieren, auszulesen und zu beschreiben. Die Daten werden als ASCII-codierte Hexadezimalzahlen übertragen und im Transponder gespeichert. Exemplarisch ist an dieser Stelle die Funktion zum Beschreiben des internen Speichers eines Transponders vom Typ TW-R20-K2 (siehe Abschnitt 3.1.3) aufgeführt. Diese Funktion heißt *RFID_schreibe* und ist ausgiebig kommentiert (grün), weshalb auf eine Erläuterung der einzelnen Zeilen hier im Text verzichtet wurde.

⁷⁷ [FEIG2015], S.23f

```

function [ ] = RFID_schreibe(rfid,snr,adr,daten)
%schreibt übergebene Daten in gewählten Transponder
%RFID-Handlevariable, snr, Startadresse, Daten (als String!) übergeben
y=double(daten); %Variable "daten" wird in ASCII gewandelt
x=fliplr(y); %spiegeln der Datenzeichen
l=length(x); %Bestimmung der Anzahl der zu speichernden Zeichen
%Bestimmung wie viele Bytes übertragen werden sollen
while mod(l,8)~=0
    x=[0 x];
    l=length(x);
end
anz_bits=20+l; %Bestimmung Anz. der benötigten Speicherbits
n=ceil(l/8); %Bestimmung Anz. der benötigten Speicherbytes
if (n+adr)>250;disp('n+adr zu hoch');return;end
%Prüfung ob Speicherkapazität des Transponders ausreicht
if n>32
%Prüfung ob Speicherkapazität des Transponders ausreicht
    disp('zu viele Daten');
else
    crc16=crc([2 0 anz_bits 255 176 36 1 snr adr n 8 x]);
    %Prüfsummenberechnung
    rfid=sende_2(rfid,[2,0,anz_bits,255,176,36,1,snr,adr,n,8,x,...
    mod(crc16,256),fix(crc16/256)]);
    %Senden der Daten
end
[rfid,x]=empfange_2(rfid,8);
%Empfangen von 8 Antwort-Bytes des Readers
if x(6)==0
%Abruf Status des Readers (Status befindet sich in Byte 6 der Antwort)
    disp('Übertragung erfolgreich');
else
    disp('Übertragung fehlgeschlagen');
end
end

```

4.1.3 Sonstige implementierte Funktionen

Neben dem RFID-Teilsystem sind auch die anderen Teilsysteme, wie in Abbildung 18 dargestellt, über Ethernet vom Steuerungscomputer aus bedienbar. Auf ähnliche Weise wie beim RFID-System ist es möglich, aus der Matlab-Programmierungsumgebung heraus Befehle über Ethernet an den jeweiligen Teilnehmer zu senden. Dabei ist zu beachten, dass für jedes Gerät im Netzwerk, das über den Ethernet-Switch (siehe Abbildung 18) verbunden ist, eine eigene IP-Adresse gewählt wird. Analog zur RFID-Kommunikation ist für den Greifer und den Roboter pro Befehl ein eigenes Skript oder eine Funktion in Matlab implementiert. Diese Funktionen und Skripte wurden schon im Zuge vorangegangener Projekte an der Hochschule in die Matlab-Programmierungsumgebung übernommen. Deshalb sollen in Tabelle 5 die – im Rahmen dieser Masterarbeit genutzten – Funktionen zur Bedienung von Roboter und Greifer lediglich genannt werden.

	Name	Funktionsbeschreibung
Roboter	<i>generate_position</i>	erstellt die Positionsvariablen, ruft Funktionen auf, die Messpositionen aus Koordinatensystem des Messbereichs in das Roboterkoordinatensystem wandeln
	<i>fahre_zu_position</i>	Verfährt Roboter zu vorher erzeugten Positionen
Greifer	<i>GetOpeningWidth</i>	Gibt die Öffnungsweite der Finger zurück
	<i>SetForceLimit</i>	Begrenzt die Greifkraft
	<i>ReleasePart</i>	Öffnet den Greifer und lässt Objekt los
	<i>GraspPart</i>	Schließt den Greifer und greift das Objekt

Tabelle 5 – Implementierte Funktionen Roboter und Greifer

Diese Funktionen werden selbstverständlich um die Funktionen zum Verbindungsaufbau und zum Beenden der Kommunikation ergänzt. Mit Hilfe dieser Funktionen konnten im Anschluss kombinierte Algorithmen programmiert werden. Einer davon ist die Funktion *GraspPart_GetWidth*. Dabei sollen die Finger mit begrenzter Greifkraft solange aufeinander zufahren bis ein Objekt gegriffen wird. Die Breite dieses Objekts wird nach Abarbeitung der Funktion im Ergebnis zurückgegeben.

Natürlich können ebenso Funktionen der einzelnen Teilsysteme miteinander kombiniert werden. Dies stellte einen Großteil des Programmieraufwands während dieser Masterarbeit dar, weshalb darauf in den folgenden Kapiteln eingegangen wird.

4.2 Ortung in der Ebene

Das verwendete RFID-System mit dem Reader ID ISC.LRM1002 und der vergleichsweise großen Antenne ID ISC.ANT310310-A wird häufig beispielsweise in Bibliotheken zur Erfassung der Klebetransponder auf den Büchern eingesetzt. Für diese Anwendung spielt die Signallaufzeit zwischen Transponder und Reader keine große Rolle. Somit hat der Reader keine Funktion, die die Signallaufzeit ausgibt. Deshalb konnte das unter Punkt 2.4 beschriebene Verfahren nicht zur Ortsbestimmung verwendet werden.

4.2.1 Sendeleistungskarte

Im Rahmen dieser Masterarbeit wurde ein Verfahren entwickelt, mit dem auch ohne Signallaufzeit eine Transponderortung möglich ist. Dazu wurde die Tatsache genutzt, dass die maximale Lesereichweite von der Sendeleistung des Readers abhängt. Diese kann, wie bereits im Abschnitt 4.1.2 erläutert, aus der MATLAB-Programmierungsumgebung eingestellt werden.

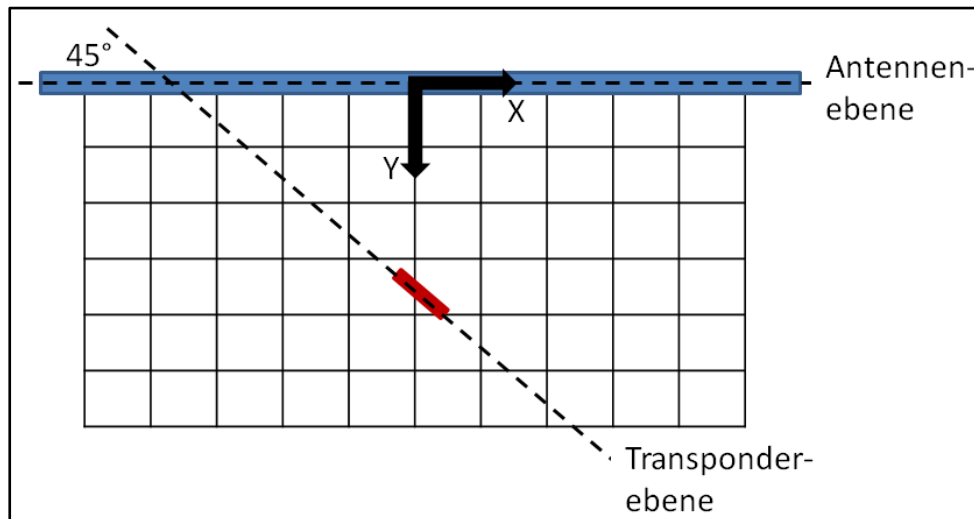


Abbildung 19 – Orientierung

Am Anfang wurde für jede Sendeleistungsstufe des Readers einzeln eine Karte erstellt, wie sie in Abbildung 20 ausschnittsweise dargestellt wurde. Dafür wurde der Messaufbau aus Abbildung 19 verwendet.

Der für den Test des Ortungsverfahrens benutzte, Transponder TW-R20-K2 (siehe 3.1.3, rot in Abbildung 19) wird an jedem Schnittpunkt eines Rasters (schwarz in Abbildung 19) platziert. Jedes einzelne Feld des Rasters ist quadratisch und hat eine Länge / Breite von 0,5 cm. Die Achsenbeschriftungen des Rasters, beziehungsweise der Matrix stehen für den Abstand des Transponders zur Mitte der RFID-Antenne (blau in Abbildung 19). Vereinfacht wurde die Matrix zweidimensional ausgeführt, weil die Ortung in der Ebene stattfinden soll und nicht im Raum. Daher konnte die Z-Koordinate außer Acht gelassen werden. Die Y-Achse des Rasters verläuft in Abbildung 19 und Abbildung 20 von oben nach unten, hat ihr Maximum bei 21 cm und den Nullpunkt in der Mitte der Antenne. Die X-Achse (quer in Abbildung 19 und Abbildung 20) misst 30 cm in beiden Richtungen von

der Antennenmitte aus gesehen. Die X-Achse ist also Teil der Antennenebene. Dies wurde in Abbildung 19 verdeutlicht. Selbstverständlich hätte man die Y-Achse analog zur X-Achse ebenfalls in beide Richtungen antragen können, dies war jedoch aufgrund des späteren Messaufbaus (Abbildung 23) nicht nötig. Ferner verlaufen die Feldlinien der Antenne, wie in Abbildung 13 zu sehen ist, symmetrisch.

Der Testtransponder wurde nun über jedem einzelnen Rasterpunkt platziert. Je nachdem bei welcher Sendeleistungsstufe der Transponder an genau dieser Stelle erkannt wurde – also eine Kommunikation möglich wurde – wurde diese Sendeleistungsstufe an der entsprechenden Stelle in die Matrix eingetragen. Auf diese Weise ist eine Karte entstanden, die zeigt welche Sendeleistung mindestens nötig ist, um mit dem Transponder, der an einer bestimmten Stelle steht, zu kommunizieren. Diese Karte beinhaltet demnach die Ziffern 1 bis 5 – diese stehen für die Sendeleistungsstufen 1 W bis 5 W – und die Ziffer 0, wenn keine Kommunikation möglich ist, also der Transponder auch bei höchster Sendeleistung nicht im Empfangsbereich ist. Dies ist in Abbildung 20 nachzuvollziehen. Zu beachten ist, dass ein Transponder, der mit niedriger Sendeleistung erkannt wird, ebenfalls mit höherer Sendeleistung erreichbar ist. Die Karte zeigt demnach die Mindest-Sendeleistung zum Erreichen der Transponder.

Wie bereits in Kapitel 3.1.2 beschrieben, ist neben der Entfernung des Transponders zur Antenne auch seine Orientierung gegenüber der Antenne entscheidend. Wenn der Transponder parallel zur Y-Achse des Rasters (siehe oben und Abbildung 19) ausgerichtet ist, wird er nicht erkannt – unabhängig von der Entfernung zur Antenne und der Sendeleistung des Readers. Aus diesem Grund wurde der Winkel zwischen der Antennenebene und der Transponderebene auf 45° festgelegt. So ist sichergestellt, dass die Antenne den Transponder erkennt, auch wenn sie um 90° gedreht wird. Zur Veranschaulichung des Messaufbaus während der Erstellung der Karte und zum besseren Verständnis der Orientierung von Transponderebene, Antennenebene und Raster dient Abbildung 19.

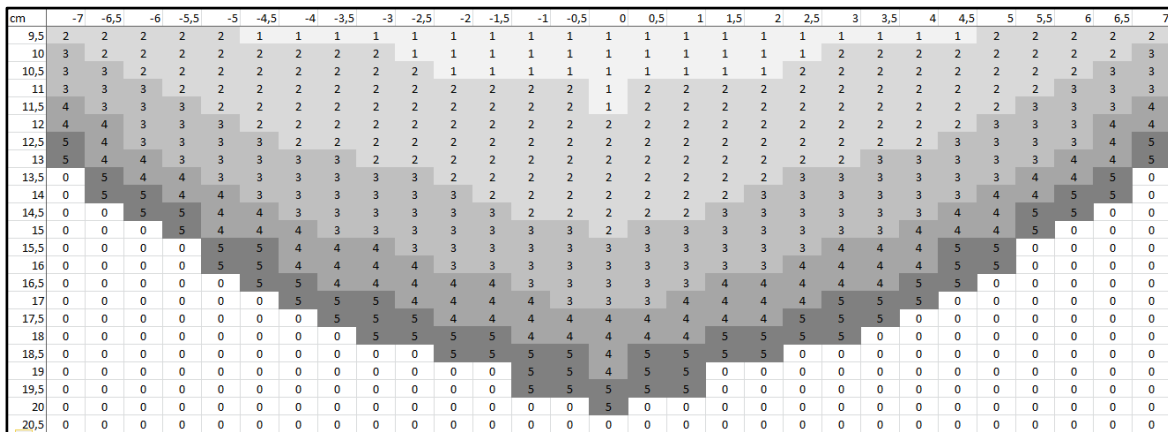
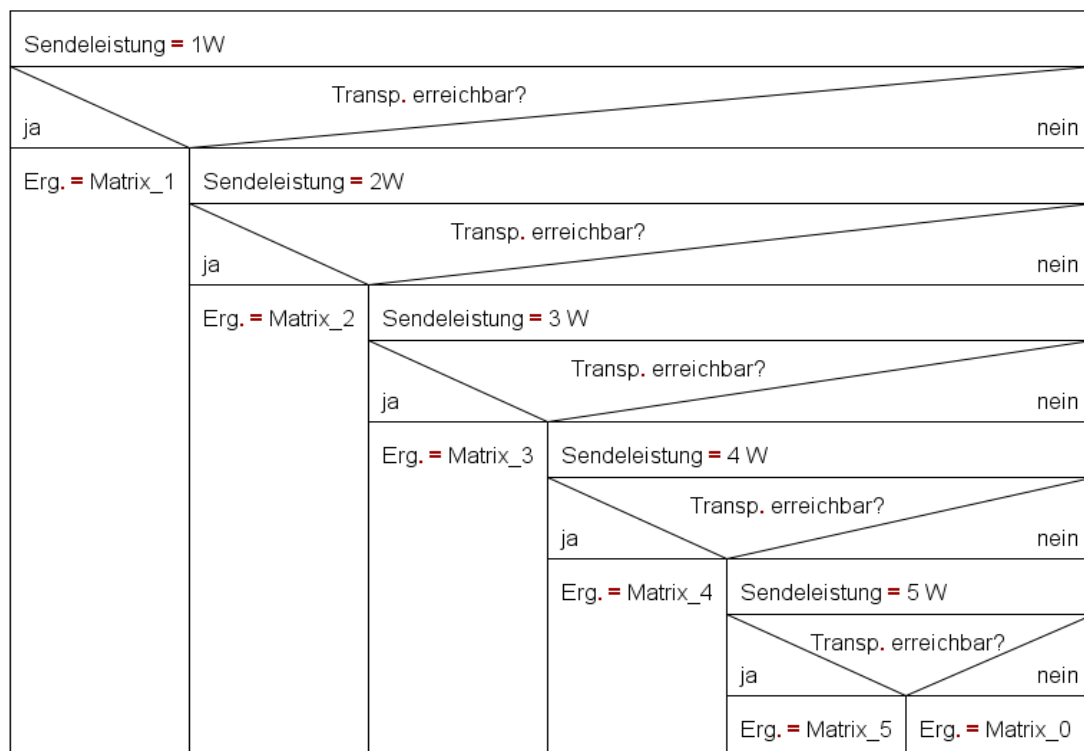


Abbildung 20 – Karte

4.2.2 Umsetzung in der Programmierung

Die erstellte Karte konnte im Anschluss in die Matlab-Programmierungsumgebung eingelesen werden und ein Programm nach dem selbsterstellten Struktogramm aus Abbildung 21 erstellt werden.

Abbildung 21 – Struktogramm Ortung über Sendeleistung⁷⁸

⁷⁸ Erstellt mit Structorizer, Version 3.19, Copyright 2009 Robert Fisch

Dieses Programm prüft für jede Sendeleistungsstufe, ob sich der gesuchte Transponder im Empfangsbereich des RFID-Readers befindet. Dabei wird bei 1 W Sendeleistung begonnen und sobald der Transponder erkannt wird, wird die Ausgabevariable mit der entsprechenden Matrix beschrieben. Diese Matrix ist stets eine von sechs Matrices, die bereits zuvor aus der Kartenmatrix auf folgende Weise erstellt wurden: Für jede Sendeleistungsstufe und für den Fall, dass sich der Transponder nicht im Empfangsbereich befindet, wurde eine gesonderte Matrix erstellt. Jede beinhaltet genau an den Stellen der Karte, an denen die Ziffer der jeweiligen Sendeleistungsstufe steht, eine 1. An allen anderen Stellen eine 0. Die Matrix für den Fall, dass der Transponder nicht im Empfangsbereich ist (*Matrix_0*), besteht ausschließlich aus Nullen. Der Algorithmus liefert also eine boolesche Matrix-Karte, die an allen möglichen Positionen, an denen sich der Transponder zum Messzeitpunkt befunden haben könnte, eine 1 ausgibt.

Natürlich kann man bei diesem Algorithmus noch nicht von Ortung sprechen, weil ganz einfach zu viele Rasterpunkte als mögliche Transponderposition ausgegeben werden. Wenn man dieses Vorgehen allerdings mehrfach von verschiedenen Messpositionen aus wiederholt, wird die Genauigkeit der Positionsangabe mit jeder Wiederholung genauer. Spätestens zu diesem Zeitpunkt der Messungen wird zur höheren Genauigkeit und besseren Reproduzierbarkeit der Einsatz des Industrieroboters in Kombination mit dem RFID-System unabkömmlich. Deshalb wurde für die folgenden Messungen die RFID-Antenne an den Industrieroboter montiert. Dies ist in Abbildung 22 zu erkennen. Die große Rahmenantenne wurde fest an den Roboter montiert. Dies schränkte natürlich die Bewegungsfreiheit des Roboters ein und musste bei der Ausrichtung der Antenne beachtet werden, um diese nicht zu beschädigen.



Abbildung 22 – RFID-Antenne an Industrieroboter

Mit diesem Aufbau wurde es möglich, von zahlreichen Messpositionen den oben und in Abbildung 21 beschriebenen Algorithmus auszuführen und somit die möglichen Positionen des Transponders einzuschränken. Der Transponder wurde zu diesem Zeitpunkt der Versuche zu dieser Masterarbeit noch provisorisch an einen Legobaustein (rot in Abbildung 22) befestigt. Dieser Baustein wurde an beliebiger Stelle des Messgebietes – einer quadratischen Legoplate (grün in Abbildung 22) – platziert.

Der Winkel zwischen Transponderebene und Antennenebene wurde auf 45° festgelegt. Welche Messpositionen genau eingenommen wurden, ist in Abbildung 23 zu sehen. Die dunkelblauen Kreuze stehen darin für den Antennenmittelpunkt während der Messung, die blauen Rechtecke symbolisieren die Antenne während der Messung, das rote Rechteck ist der Transponder und das grüne Feld stellt das Messgebiet – die Legoplate – dar. Außerdem wurde nochmals das – schon aus Abbildung 19 bekannte – Raster angedeutet. In dieser abstrahierten Darstellung hat jedes Feld des Rasters allerdings eine Breite / Länge von 2,5 cm. Benachbarte Messpunkte sind also 2,5 cm voneinander entfernt. Selbstverständlich können beliebig viele weitere Messpunkte hinzugefügt werden. Der Aufwand würde zugunsten der Genauigkeit ansteigen. Für die Erprobung des Algorithmus waren die

10 – in Abbildung 23 (dunkelblaue Kreuze) eingezeichneten – Messpunkte jedoch ausreichend.

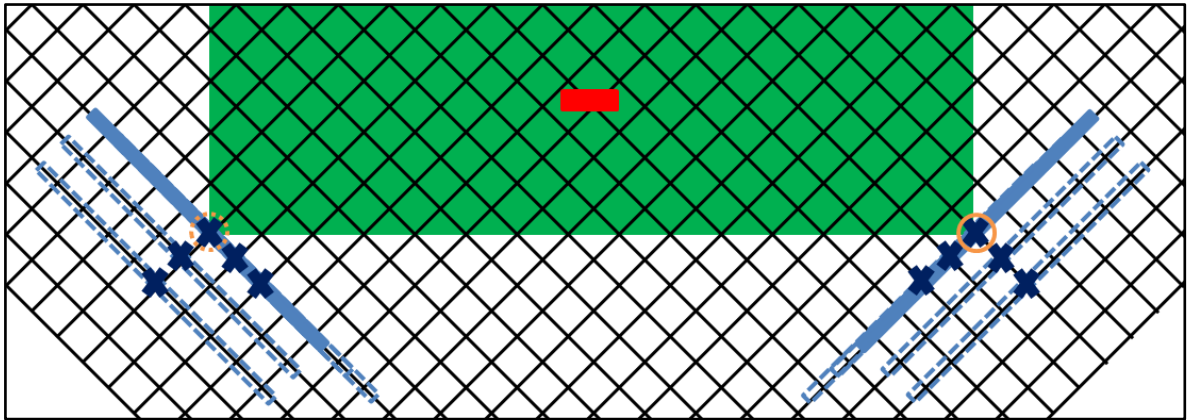


Abbildung 23 – Raster & Messpunkte

Wie bereits erwähnt, wird an jedem Messpunkt der in Abbildung 21 dargestellte Algorithmus ausgeführt. Das bedeutet, dass an jedem Messpunkt eine Matrix gespeichert wird. Die Menge der Daten – der Matrices – steigt also mit der Anzahl der Messpunkte. Zu beachten ist, dass sich jede Matrix auf den aktuellen Messpunkt bezieht. Die an jedem Messpunkt gespeicherte Matrix enthält die möglichen Transponderpositionen, beziehungsweise die denkbaren Entfernungen des Transponders, in Bezug auf den aktuellen Messpunkt. Wenn man nun aus allen gespeicherten Matrices der verschiedenen Messpunkte auf eine Transponderposition schließen möchte, muss man alle Matrices auf einen Bezugspunkt, ein Koordinatensystem umrechnen. In diesem Fall wurde als Bezugspunkt der Messpunkt an der rechten unteren Ecke des Messgebietes / der Legoplatte (siehe Abbildung 23, oranger Kreis) gewählt. Die Umrechnung der anderen Matrices erfolgt folgendermaßen: Wenn der Messpunkt beispielsweise um 2,5 cm nach links verschoben wird (vom Bezugspunkt aus gesehen auf seiner Antennenebene), so muss die Matrix des zweiten Messpunktes in die entgegengesetzte Richtung um den gleichen Betrag verschoben werden – in diesem Fall also um 2,5 cm, oder 5 Felder des Rasters nach rechts. Dies geschieht im Beispiel rechnerisch, indem man 5 Spalten Nullen vor die ausgegebene Matrix (*c13*) anfügt und anschließend die letzten 5 Spalten der Matrix (*c13*) löscht. Der entsprechende Programmcode sieht folgendermaßen aus:

```
c13=zeros(41,5),c13];  
c13(:,122:126)=[];
```

Die Umrechnung der restlichen Matrices der Messpunkte an dieser Ecke des Messgebietes erfolgt analog dazu. Bei den Messpunkten die unterhalb des ersten Messpunktes liegen müssen selbstverständlich statt Spalten Zeilen angefügt beziehungsweise gelöscht werden. Die Matrices der Messpunkte in der Nähe der linken Ecke des Messgebietes werden vorerst auf die Matrix des Messpunktes an der linken Ecke des Messgebietes (gestrichelter oranger Kreis in Abbildung 23) umgerechnet. Nach den erfolgten Umrechnungen werden jeweils die beiden Gruppen von Matrices summiert, sodass nun zwei Matrices übrigblieben, welche – je nachdem von wie vielen Messpunkten aus der Transponder erkannt wurde – Zahlen von 0 bis 5 enthalten. Bevor die übrigen beiden Matrices addiert werden können, muss die Matrix des linken Bezugspunktes in die Matrix des rechten Bezugspunktes umgerechnet werden. Dazu muss die Matrix des linken Bezugspunktes erst um 90° gedreht werden, was rechnerisch dem Transponieren der Matrix entspricht. Anschließend kann die Matrix, wie oben beschrieben, in den (rechten) Hauptbezugspunkt (oranger, durchgezogener Kreis in Abbildung 23) verschoben werden. Zuletzt werden die beiden Matrices addiert. Das Resultat ist eine einzige Matrix. Erwartet wurde, dass sich das Maximum aller Zahlen der Matrix genau an der Stelle, die die Position des Transponders repräsentiert, befindet. Diese Erwartung wurde wegen Messungenauigkeiten und äußeren Einflüssen nicht bei jedem Messvorgang erfüllt. Abhilfe könnte die Erhöhung der Anzahl von Messpositionen pro Messvorgang schaffen.

4.3 Intelligentes Greifen

Im Rahmen dieser Masterarbeit sollte eine weitere Funktion implementiert werden, die die Kombination der Teilsysteme RFID und Industrieroboter erfordert. Eine grundlegende Anwendung von Industrierobotern ist das Greifen von Gegenständen. Der in Abschnitt 3.2.1 beschriebene Greifer WSG50 ist dazu bestens geeignet. Insbesondere bei empfindlichen, fragilen Gegenständen sind die einstellbaren Faktoren Greifgeschwindigkeit und Greifkraft von essentieller Bedeutung. Wenn nun die zu greifenden Gegenstände unbekannte Abmessung haben, sollte sowohl die Greifkraft als auch die Greifgeschwindigkeit begrenzt werden, um die Gegen-

stände nicht zu beschädigen. Dies hat jedoch zur Folge, dass der Greifprozess – je nach Abmessung und Öffnungsweite des Greifers – vergleichsweise lange Zeit in Anspruch nehmen kann. Zur Vermeidung dieser Problematik kann die RFID-Technik genutzt werden. Sie ermöglicht, dass Informationen wie Abmessungen, Schwerpunkt und Empfindlichkeit des zu greifenden Gegenstandes gespeichert werden können. Dazu muss jeder Gegenstand mit einem Transponder ausgestattet werden, welcher bei einem Vorgang, der zeitlich vor dem eigentlichen Greifen stattfindet, beschrieben wird. Vor dem Greifen können demnach die Abmessungen aus dem Transponder im Gegenstand ausgelesen werden. So kann der Greifprozess beschleunigt werden, wenn mit voller Greifgeschwindigkeit bis kurz vor den Gegenstand herangefahren wird und anschließend sanft der Kontakt zum Gegenstand hergestellt wird. Die Informationen über Schwerpunkt und Empfindlichkeit des Gegenstandes können bei der Wahl der Greiffaktoren ebenfalls nützlich sein.

4.3.1 Hardware-Umsetzung



Abbildung 24 – Antennen an Greifer

Für diese Anwendung muss die RFID-Technik mit einem Greifer kombiniert werden. Deshalb wurde hier, anders als bei dem Aufbau für die Transponderortung – siehe Abbildung 22 und Abschnitt 4.2 – der Greifer WSG 50 und eine RFID-Antenne gleichzeitig am Roboter montiert. Die Rahmenantenne ID

ISC.ANT310310-A erwies sich wegen ihrer Größe als ungeeignet für diese Aufgabe. Deshalb kamen die deutlich kleineren Platinen-Antennen, die am Ende des Kapitels 3.1.2 beschrieben wurden, zum Einsatz. Die Antennen wurden direkt an den Greifer montiert, weil die Reichweiten der vergleichsweise kleinen Antennen begrenzt sind. Der Einfluss von Metall auf die Funktion des RFID-Systems und die Reichweite des Feldes spielte hier eine große Rolle. Abbildung 24 zeigt, wie die beiden Platinen-Antennen an den Greifer montiert wurden. Das Befestigungsmaterial für die Montage der größeren der beiden Antennen (ID ISC.ANT100/100, links in Abbildung 24) besteht ausschließlich aus Kunststoff. Außerdem wurde darauf geachtet, dass der Arbeitsbereich des Greifers so wenig wie möglich beschränkt wird. Die Antenne konnte nicht direkt an den Finger des Greifers montiert werden, weil dieser aus Metall besteht und somit das von der Antenne ausgesendete magnetische Feld stören würde. Da die Leiterschleife der Antenne ganz außen in der 10 mal 10 Zentimeter großen Platine eingelassen ist, verläuft ihr magnetisches Feld auch um den Finger des Greifers herum und wird zwar beeinflusst, aber nicht vollständig blockiert. Dies war der Fall bei der kleineren Platinen-Antenne (ID ISC.ANT40/30, rechts in Abbildung 24). Deshalb wurde eigens zur Montage dieser Antenne der metallene Finger des Greifers in einem 3D-Drucker nachgebaut. Durch diesen Kunststofffinger (rot auf der linken Seite von Abbildung 24) wurde der Betrieb mit der kleineren, für diese Anwendung besser geeigneten Antenne möglich.

Nach der Montage konnte der Empfangsbereich dieser neuen RFID-Systeme ermittelt werden. So zeigt Abbildung 25 exemplarisch den approximierten Empfangsbereich des auf der linken Seite in Abbildung 24 dargestellten Aufbaus. Auf Basis dessen konnte die weitere Hardware und Software zur Umsetzung der Aufgabe des intelligenten Greifers installiert und konfiguriert werden.

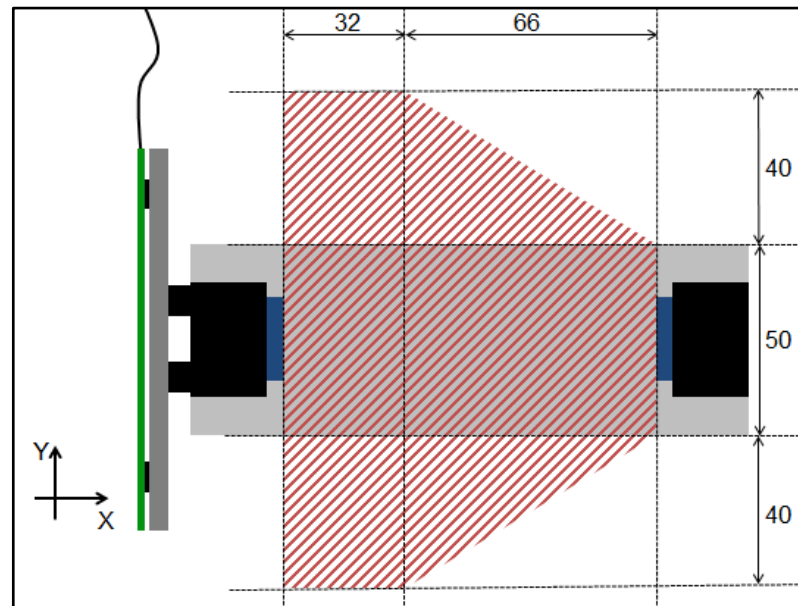


Abbildung 25 – RFID-Empfangsbereich mit ID ISC.ANT100/100 (Maße in mm)

Der Empfangsbereich des magnetischen Feldes, das vom RFID-System mit der kleineren Antenne ausgestrahlt wird, ist besonders in Y-Richtung nur unwesentlich schmaler als jener in Abbildung 25. Aus diesem Grund kam der Aufbau mit der kleineren Antenne und dem roten Kunststoff-Finger verstärkt zum Einsatz, weil dieser Aufbau die Funktion des Greifers nicht beeinträchtigt.

Für die Erprobung des intelligenten Greifers wurden eigens fünf Bausteine unterschiedlicher Breiten angefertigt. Der Fuß jedes Bausteins wurde einem typischen 4x2 reihigen Legobaustein nachempfunden, sodass die Bausteine sicher und definiert auf einer Legoplatte (wie in Abbildung 26) platziert werden konnten. Außerdem wurden die Bausteine mit jeweils 2 Schlitz, die zueinander um 90° verdreht sind, versehen. Jene Aussparungen ermöglichten das Bestücken mit je einem Transponder vom Typ TW-R20-K2 in definierter Position und Orientierung. Dies gewährleistet, dass unabhängig von der Orientierung der Antenne zum Baustein mindestens ein Transponder des Bausteins erreichbar ist – unter der Voraussetzung, dass sich der Baustein in geeigneter Entfernung befindet. Wie in Abbildung 26 zu sehen ist, wurden die Bausteine in einer Linie angeordnet um im nächsten Schritt die Breite jedes einzelnen Bausteins, wie bereits erwähnt, in den integrierten Transponder zu schreiben. Voraussetzung dafür war, dass während der Mes-

sung der Breite der Bausteine stets nur der in diesem Baustein, in dieser Orientierung integrierte Transponder erreichbar war. Deshalb durfte der Abstand der Bausteine zueinander nicht kleiner als 6,5 cm ($40\text{mm} + (50\text{mm})/2$ siehe Abbildung 25) gewählt werden. Bei dem Aufbau aus Abbildung 26 beträgt der Abstand der Bausteine (gemessen von der Mitte des einen zur Mitte des nächsten Bausteins) 8 cm.

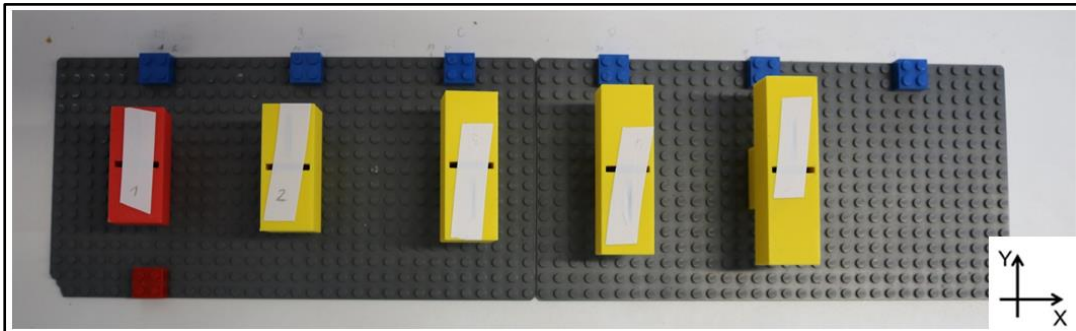


Abbildung 26 – Bausteine mit integrierten Transpondern

4.3.2 Software-Umsetzung

Wie bereits unter Punkt 4.3 erwähnt, soll das Greifen in zwei Stufen ablaufen. Im ersten Schritt sollen die Bausteine vorsichtig, langsam, gegriffen werden und deren dabei ermittelte Abmessung in dem Transponder gespeichert werden. Die entsprechende Funktion beherrscht der Greifer, beziehungsweise dessen Software, nicht von Haus aus. Allerdings hat der Hersteller des Greifers selbstverständlich die Funktionen zum Zugreifen und zur Ausgabe der aktuellen Öffnungsweite der Finger implementiert. Diese Funktionen lagen, wie die Funktionen zur Robotersteuerung, durch vorangegangene Projekte an der Hochschule bereits in der Matlab-Programmierung vor. Zur Realisierung der ersten Stufe des intelligenten Greifens reichten demnach folgende Programmzeilen aus:

```
WSG_SetForceLimit(t, 50);
WSG_GraspPart(t, 0, 50);
width = WSG_GetOpeningWidth(t);
w=num2str(width);
disp(['Breite: ' w]);
RFID_schreibe(rfid,snr,0,w,2);
WSG_ReleasePart(t, 100, 100);
```

Auf Programmkommentare wurde an dieser Stelle verzichtet, weil jede Zeile im Folgenden erklärt wird. Dieser Programmausschnitt wird ausgeführt, sobald der Roboter den montierten Greifer genau über dem Objekt positioniert hat und sich somit das Objekt genau in der Mitte der beiden Finger des Greifers befindet. Zuerst wird die Greifkraft des Greifers begrenzt, um das Objekt nicht zu beschädigen – hier exemplarisch auf 50 N. Anschließend wird der Greifer aufgefordert, einen Gegenstand mit der Breite von 0 cm zu greifen. Wenn sich allerdings ein Baustein zwischen den Fingern des Greifers befindet, kann natürlich nicht auf 0 cm Öffnungsweite zugegriffen werden – der Greifer registriert, dass sich ein Objekt zwischen den Fingern befindet, bricht das Zufahren ab und greift somit das Objekt mit der unbekannten Breite. Diese wird – nachdem der Greifvorgang beendet wurde – in der dritten Zeile des oben aufgeführten Codes ausgelesen und in der numerischen Variable „*width*“ gespeichert. Diese wird in der nächsten Zeile in eine Variable vom Typ „string“ umgewandelt und in der Variable *w* gespeichert. In dieser Form kann die Öffnungsweite im Matlab-Kommandofenster ausgegeben werden (Zeile 5) und im RFID-Transponder gespeichert werden. Dazu wird die Funktion „*RFID_schreibe*“ benutzt, welche bereits unter Punkt 4.1.2 erläutert wurde. Als letztes soll sich der Greifer wieder maximal (100 mm) öffnen, um den nächsten Baustein abarbeiten zu können.

Die zweite Stufe des intelligenten Greifens besteht darin, dass vor dem eigentlichen zugreifen durch den Greifer, die Breite des Bausteins ausgelesen wird. Mit dieser Information können die Finger mit voller Geschwindigkeit an das Objekt heranfahren, um dann kurz vor dem Kontakt mit dem Objekt abzubremsen und sanft zuzugreifen. Dies wäre ohne die RFID-Technik nur durch Zwischenspeicherung der Objektabmessungen in der Matlab-Programmierungsumgebung möglich. Dank der integrierten Transponder sind aber auch Positionsveränderungen zwischen den beiden Stufen möglich. Außerdem gilt, dass nachdem der im Baustein integrierte Transponder einmal beschrieben wurde, kann er beliebig oft und in beliebig großen Zeitabständen schnell beziehungsweise „intelligent“ gegriffen werden. Der Algorithmus für die zweite Stufe des intelligenten Greifens ist, Dank der bereits implementierten Funktionen des RFID-Systems und des Greifers, ähnlich einfach wie der Algorithmus der ersten Stufe. Die Breite des Bausteins wird aus

dem Transponder ausgelesen und nach einer Variablenumwandlung an die Greif-Funktion weitergegeben.

4.3.3 Eindimensionale Transponderortung

Mit dem in Abschnitt 4.3.1 erklärten und in Abbildung 24 rechts, sowie Abbildung 26 dargestellten Aufbau konnte sich erneut der Transponderortung zugewandt werden. Ziel sollte es sein, dass ein Baustein, der mit einem Transponder ausgestattet wurde, gegriffen werden kann, ohne dass die Position des Bausteins bekannt ist. Dazu mussten einige Beschränkungen vorgenommen werden: Als Messgebiet, also der Bereich in dem sich der Baustein befinden darf, wurde die graue Legofläche (siehe Abbildung 26) zwischen dem ersten (links) und letzten (rechts) blauen Legosteine gewählt. Durch die Noppen auf der Legoplatte und die typische Form eines Legosteins als Fuß der Bausteine ergab die Wahl des Messgebietes eine endliche Anzahl möglicher Bausteinpositionen. Zusätzlich wurde die Y-Koordinate auf die Mitte der grauen Legoplatte / des Messgebietes festgelegt. Diese Einschränkung musste vorgenommen werden, weil nur so sichergestellt werden konnte, dass beim Abfahren der Messstrecke weder Greifer, noch Baustein beschädigt werden. Die möglichen Positionen des Bausteins durften sich also nur in einer Dimension unterscheiden – es durfte sich nur die X-Koordinate verändern. Daraus ergeben sich im oben genannten Messgebiet mit der Breite (Ausdehnung in X-Richtung) von 41,6 cm – beziehungsweise 52 Noppen – 51 mögliche Bausteinpositionen.

Zur Transponderortung in diesem eindimensionalen Gebiet wurden zwei verschiedene Algorithmen entwickelt, welche beide ihre Vor- und Nachteile haben: Im ersten Verfahren positioniert der Roboter – wie in Abbildung 27 dargestellt – den voll geöffneten Greifer (mit montierter RFID-Antenne) nacheinander über jeder möglichen Position. An jeder Position wird zuerst die Transponderanzahl im Empfangsbereich ermittelt. Wenn sich mindestens ein Transponder meldet, werden alle Seriennummern der Transponder im Empfangsbereich ausgelesen und mit der Seriennummer des gesuchten Transponders verglichen. Der Greifer muss dazu solange über der aktuellen Messposition verweilen bis die Seriennummern ausgelesen sind. Würde er sich nach der Bestimmung der Transponderanzahl weiterbe-

wegen, könnte es passieren, dass sich dadurch die Anzahl verändert und somit würde die Funktion zum Auslesen der Seriennummer einen Fehler produzieren. Daraus ergibt sich folgende Erkenntnis: Dieses erste Verfahren erwies sich als vergleichsweise langsam, aber die Transponderanzahl im Messgebiet ist beliebig. Darin besteht der Unterschied zum zweiten Verfahren. Hier verfährt der Roboter den voll geöffneten Greifer kontinuierlich, mit gleichbleibender Geschwindigkeit entlang des Messgebietes vom ersten blauen Legostein bis zum letzten (siehe Abbildung 27). Dabei ist es allerdings nicht möglich die Seriennummer auszulesen, weil sich, wie oben beschrieben, während des Auslesens die Transponderanzahl verändern kann. Aus diesem Grund darf sich bei diesem Verfahren nur ein Transponder im Messbereich (und in dessen direkter Umgebung) befinden.

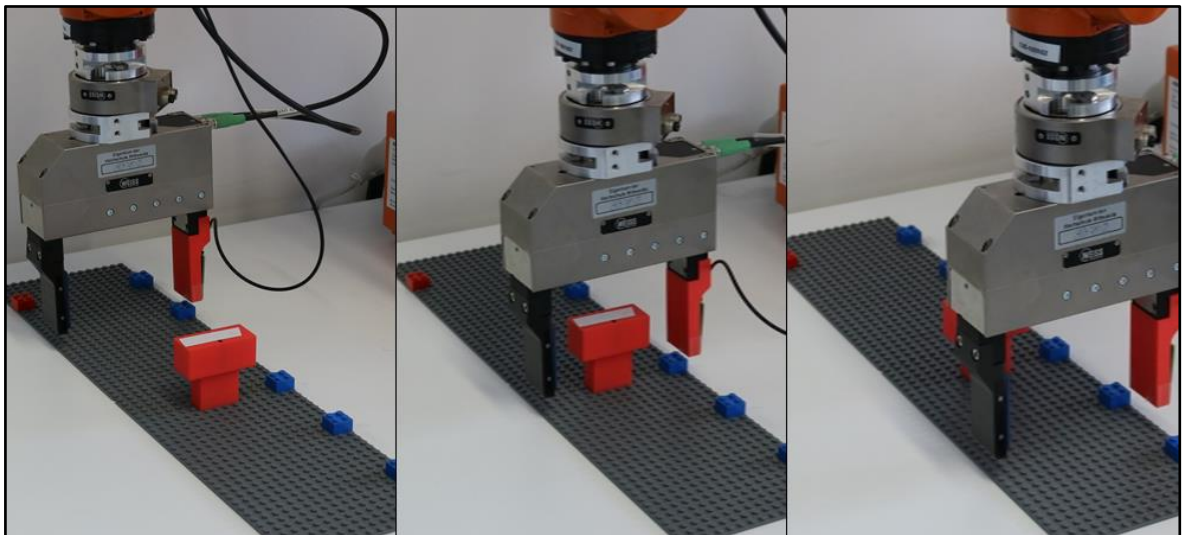


Abbildung 27 – Eindimensionale Ortung

Neben dem Messgebiet haben die beiden Verfahren die Methode, mit der aus den Messdaten auf die Bausteinposition geschlossen wird, gemeinsam. Bei jedem Messvorgang wird eine Spalte zu einer Tabelle hinzugefügt. In die erste Zeile der Tabelle wird stets die aktuelle X-Koordinate geschrieben. Je nachdem ob an der aktuellen Position das Signal des (gesuchten) Transponders empfangen wird, wird in die zweite Zeile eine Null (nicht erreichbar) oder eine Eins (erreichbar) geschrieben. Nach Abschluss des Messvorgangs wird der Mittelwert der X-Koordinaten, an denen das Transpondersignal empfangen wurde (Einsen in der Tabelle) gebildet. Diese Koordinate entspricht dann der Bausteinposition. Da dieser Algorithmus nicht funktioniert, wenn sich der Baustein auf den ersten oder letz-

ten Positionen des Messbereiches befindet, musste der Algorithmus für diese Spezialfälle noch optimiert werden. Dazu wurden diese Fälle fest einprogrammiert.

4.4 Sortierung

Durch das Bestücken der Bausteine mit RFID-Transpondern und die Kombination des Greifers WSG 50 mit einer RFID-Antenne boten sich noch weitere Aufgaben, die dieses kombinierte System übernehmen kann. Umgesetzt wurde das Sortieren der Bausteine nach der Größe – eine komplexe Aufgabe, die alle Funktionen miteinander kombiniert:

- Verfahren des am Roboter montierten Greifers
- Ausmessen der Bausteine
- Beschreiben der Transponder
- Auslesen der Transponderdaten
- Intelligentes Greifen der Bausteine
- Verschieben, Sortieren der Bausteine

Bevor die Umsetzung des gesamten Sortiervorgangs erläutert wird, soll unter Punkt 4.4.1 auf die zugrundeliegenden Algorithmen des reinen Sortierens eingegangen werden – auch dafür wurden verschiedene Herangehensweisen und Algorithmen implementiert.

4.4.1 Sortialgorithmen

Das Sortieren ist eines der Grundproblematiken der Informatik. Es existieren deshalb zahlreiche Algorithmen, die dieses Problem lösen können. Die Eingangsvariable ist stets eine (unsortierte) Folge / eindimensionale Matrix von Zahlen. Der Sortialgorithmus ordnet diese Zahlenfolge. Die Algorithmen unterscheiden sich im Endeffekt nur um die Dauer, beziehungsweise die Anzahl Züge bis aus der unsortierten Folge die sortierte, ausgegebene Folge entsteht. Zwei einfache Algo-

rithmen sind Bubblesort⁷⁹ und Selectionsort⁸⁰, sie werden in Tabelle 6 verglichen werden.

Name	Beschreibung
Bubblesort	<ul style="list-style-type: none"> • Vergleich benachbarter Elemente, beginnend beim ersten und zweiten, dann das zweite und dritte, usw. • falls die erste Zahl größer ist als die zweite, werden die beiden vertauscht • im ersten Durchgang wandert also das größte Element an die letzte Stelle (wie eine aufsteigende Blase – engl. „Bubble“), im zweiten das zweitgrößte an die vorletzte Stelle, usw. • Die Zahlenfolge wird also $(n-1)$ mal durchlaufen. (n=Länge der Zahlenfolge)
Selectionsort	<ul style="list-style-type: none"> • Die Zahlenfolge wird nach dem kleinsten Element durchsucht. Dieses Element wird mit dem ersten Element vertauscht. • Anschließend wird das zweitkleinste gesucht und an die zweite Stelle verschoben, usw. • Auch hier wird die Zahlenfolge $(n-1)$ mal durchlaufen

Tabelle 6 – Bubblesort und Selectionsort⁸¹

Beide Verfahren wurden noch etwas verbessert: Sobald in einem Durchgang keine Vertauschung mehr stattfanden, wird der Algorithmus beendet. Selectionsort hat sich nach einigen Testläufen als besseres, schnelleres Verfahren herausgestellt. Doch beide Sortieralgorithmen haben einen Nachteil. Für jeden Tauschvor-

⁷⁹ [Lang2017a]

⁸⁰ [Lang2017b]

⁸¹ [Lang2017a]; [Lang2017b]

gang zweier Bausteine sind drei Greifvorgänge nötig. Der erste Baustein wird angehoben und auf der Austauschposition abgestellt. Dann wird der zweite Baustein auf die bisherige Position des ersten Bausteins gesetzt. Zuletzt wird der erste Baustein von der Ausweichposition auf die bisherige Position des zweiten Bausteins verfahren. Deshalb wurde der Selectionsort-Algorithmus verbessert: Es gibt keine feste Ausweichposition mehr. Das Element, das an der ersten Stelle steht, wird auf die aktuell leere Stelle verschoben, das kleinste Element wird an die erste Stelle verschoben. Die Verbesserung besteht nun darin, dass die Stelle an der das kleinste Element zuvor stand, jetzt die neue Ausweichposition ist. Somit sind nur zwei Greifvorgänge pro Austauschvorgang nötig. Tabelle 7 zeigt einen exemplarischen Vergleich der drei Sortieralgorithmen. Die Ausweichposition ist darin grün dargestellt. In diesem Beispiel ist besonders der Vorteil von Selectionsort gegenüber Bubblesort zu sehen. Natürlich ist Selectionsort nicht immer so viel besser als Bubblesort. Die verbesserte Version von Selectionsort mit variabler Ausweichposition ist jedoch stets mindestens genauso gut / schnell wie die anderen Algorithmen.

Greifvorgang			0	1	2	3	4	5	6	7	8	9	10	11	12	...	21
Bubblesort	Position / Konstellation	A	4	4	4	4	4	4	4	4	4	4		1	1	...	1
		B	5		1	1	1	1	1	1	1	1	1		4	...	2
		C	1	1		5		3	3	3	3	3	3	3	3	...	3
		D	3	3	3	3	3		5		2	2	2	2	2	...	4
		E	2	2	2	2	2	2	2	2		5	5	5	5	...	5
		F		5	5		5	5		5	5		4	4		...	
Selectionsort	Position / Konstellation	A	4		1	1	1	1	1	1	1	1	-	-	-	-	-
		B	5	5	5	5		2	2	2	2	2	-	-	-	-	-
		C	1	1		4	4	4	4		3	3	-	-	-	-	-
		D	3	3	3	3	3	3	3	3		4	-	-	-	-	-
		E	2	2	2	2	2		5	5	5	5	-	-	-	-	-
		F		4	4		5	5		4	4		-	-	-	-	-
Selectionsort verbessert	Position / Konstellation	A	4		1	1	1	1	1	1	-	-	-	-	-	-	-
		B	5	5	5		2	2	2	2	-	-	-	-	-	-	-
		C	1	1		5	5		3	3	-	-	-	-	-	-	-
		D	3	3	3	3	3		4	4	-	-	-	-	-	-	-
		E	2	2	2	2		5	5	5	-	-	-	-	-	-	-
		F		4	4	4	4	4	4		-	-	-	-	-	-	-

Tabelle 7 – Beispiel Sortieralgorithmen

Es wurde bei diesem Vergleich nicht die Anzahl von Vertauschungen, die nötig sind bis die Sortierte Folge vorliegt, als Vergleichsparameter herangezogen, sondern die Anzahl von Greifvorgängen, die das System aus Greifer und Roboter durchführen muss. Dieser Parameter ist in unserem Fall der entscheidende, weil das Vertauschen der Bausteine durch den Roboter die meiste Zeit in Anspruch nimmt.

4.4.2 Umsetzung der Sortierung

Als Messbereich für diese Aufgabe wurde erneut der Aufbau, der schon vom intelligenten Greifen (Abschnitt 4.3) bekannt ist, genutzt. Die Bausteine sollten sich wie in Abbildung 26 stets auf Höhe der fünf ersten (linken) blauen Legosteine – in beliebiger Konstellation – befinden. Die somit freie Position unter dem sechsten blauen Legostein bildete die Ausweichposition.

Voraussetzung für einen Sortieralgorithmus ist, dass alle Bausteine im Messbereich – wie beim intelligenten Greifen mit dem Greifer ausgemessen werden. Diese Abmessung wird im integrierten Baustein gespeichert. Einmal beschrieben, können die Bausteinabmessungen beliebig oft aus den Transpondern ausgelesen werden. Der folgende Ausschnitt ist das Ende des Scripts, welches stets vor dem Sortieralgorithmus aufgerufen wird. Es liest die Abmessungen jedes Bausteins aus (In diesem Ausschnitt ist nur das Auslesen des fünften und letzten Bausteins aufgeführt. Das Auslesen der ersten vier Bausteine erfolgt analog) und speichert sie in der Variable / dem Vektor `breiten`.

```

%% Position 5 (=am fünften blauen Legostein)
Target_Position=Vorpos_(5,:);      %Roboter verfährt Greifer
fahre_zu_Position;                  %über Position 5
Target_Position=Ablagepos_(5,:);    %Roboter verfährt Greifer
fahre_zu_Position;                  %an Position 5

anz=RFID_anzahl(rfid); %Auslesen der Transponderanzahl
if anz==1 %wenn ein Transp. erreichbar
    snr1=RFID_snr(rfid,1); %Auslesen der Seriennummer
    breite5=RFID lese(rfid,snr5{1},0,1,2); %Breite auslesen
    breite5=str2double(char(fliplr(breite5{1})));
    %Variable breite1 umrechnen
else %wenn kein Transponder erreichbar
    disp('kein Baustein auf Position 1'); %Anzeige
    breite5=0; %breite5 auf null setzen
end

%% Variablendeklaration "breiten"
breiten=[breite1 breite2 breite3 breite4 breite5];

```

Auf eine Beschreibung jeder einzelnen Zeile des Codes wird an dieser Stelle verzichtet, weil jeder Befehl kommentiert (grün) wurde. Die Variable `breiten` enthält nach Aufruf des Scripts fünf unsortierte Werte und wird an einen der unter Punkt 4.4.1 erläuterten Sortieralgorithmen übergeben. Die Programmcodes der Sortieralgorithmen wurden aus den Algorithmen, die man in der Literatur findet, abgewandelt, beziehungsweise in die Matlab-Programmiersprache umgeschrieben. Aus diesem Grund und weil die Codes relativ lang sind, sind sie im Anhang (Anlage 1) dieser Masterarbeit zu finden.

Gemeinsam haben alle, dass sie während der Abarbeitung des Scripts (mehrfach) eine Form der Funktion `tausche` aufrufen. An diese Funktion müssen neben den beiden Positionen, deren Bausteine getauscht werden sollen, auch die aktuelle Variable `breiten` sowie die Handle-Variablen von Roboter und Greifer übergeben werden. Die Funktion beinhaltet die Befehle zum Tausch der beiden Bausteine auf den übergebenen Funktionen, beziehungsweise die Befehle dafür, dass der Baustein auf der ersten übergebenen Position auf die zweite transportiert wird. Dazu gehören also Anweisungen an den Roboter, um den Greifer zu verfahren und Befehle an den Greifer. Es existieren mehrere Formen der Funktion `tausche`, weil zwischen Austausch mit fester Ausweichposition (bei Bubblesort und Selectionsort) und mit variabler Ausweichposition (verbesserter Selectionsort) unterschieden

wird. Außerdem wurden verschiedene Versionen programmiert für die Fälle, dass sich auf einer der beiden Positionen kein Baustein befindet.

Abbildung 28 zeigt den Messbereich während der Ausführung des Sortieralgorithmus. Vom Einlesen der Bausteinbreiten (oben links) bis zur sortierten Bausteinreihe (unten links). Bei der Aufnahme dieser Bilder kam der verbesserte Selectionsort Sortieralgorithmus mit variabler Ausweichposition zum Einsatz. So wurde also zuerst der Baustein von Position 1 auf die Ausweichposition gebracht (2. und 3. Bild oben). Danach wurde der kleinste, rote Baustein auf die erste Position platziert (4. Bild oben). Der Baustein, der auf die Ausweichposition gebracht wurde blieb vorerst dort und die Position, an der der rote Baustein stand, wurde somit zur neuen Ausweichposition. Auf diese Weise wurde so lange fortgefahren, bis der sortierte Zustand – wie im Bild unten rechts – vorlag.

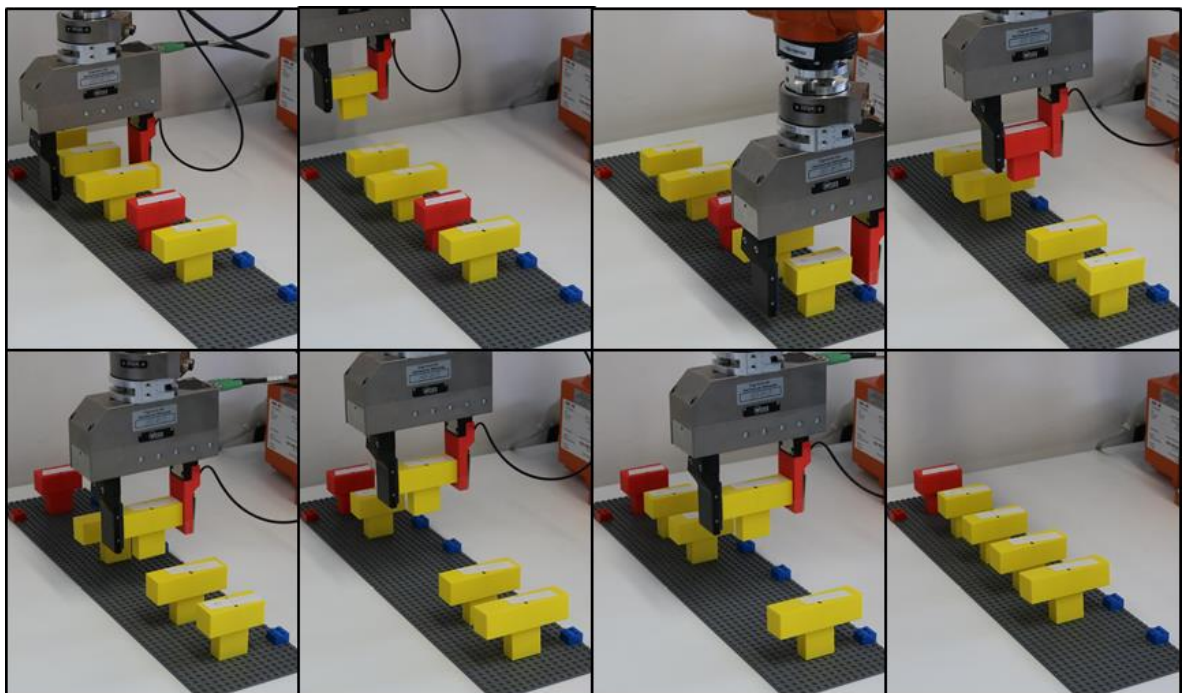


Abbildung 28 – Bildfolge Sortieren

Auf dem dieser Arbeit beiliegenden und in Anlage 2 dokumentierten Datenträger findet man ein Video, das den Messbereich während der Ausführung des Sortieralgorithmus zeigt. Auf dem Datenträger sind weitere Videos und Bilder zu den realisierten Funktionen gespeichert. Diese könnten Lehrmaterialien für die Studenten der HS Mittweida ergänzen.

5 Fazit und Ausblick

In der industriellen Automation und Automatisierungstechnik ist es stets von Bedeutung, vor Projektstart Kenntnis über die zu automatisierenden / programmierenden Prozesse oder Geräte zu haben. So musste auch für das Projekt der Kombination von RFID-Technologie mit einem Industrieroboter fundiertes Grundwissen geschaffen werden. Deshalb wurden zum Auftakt dieser Masterarbeit viele notwendige Grundlagen zum Verständnis der RFID-Technologie gelegt. Ferner wurden auf dieser Basis einige Anwendungsmöglichkeiten der RFID-Technologie aufgezeigt. Auf Grund der zahlreichen Möglichkeiten der Kombination und Konfiguration der einzelnen RFID-Komponenten sind die Anwendungsgebiete ebenfalls breit gefächert. Es wurde sich deshalb bewusst auf Anwendungen aus der industriellen Automation konzentriert, um den Fokus auf dem Thema und der Zielsetzung der Masterarbeit zu belassen. Die RFID-Technik findet bereits breite Anwendung auf diesem Gebiet. So kommt sie beispielsweise zum Einsatz, um im Automobilbau – bereits im Rohbau – sämtliche Informationen zum Fahrzeug auf dem RFID-Transponder zu speichern. Dieser Transponder ist direkt auf dem Fahrzeug montiert und liefert zu jedem Zeitpunkt der Montage Informationen über die Ausstattung und Konfiguration des Automobils. Auf diese Weise erlangt ein Mitarbeiter oder ein Roboter Kenntnis über die nötigen nächsten Arbeitsschritte. Ähnliche Anwendungen existieren auch für andere Industriezweige.

Die meiste Zeit während der Erstellung der Masterarbeit nahmen die Programmierungen in Anspruch. Aus diesem Grund bildet die Beschreibung dieses Themas in Abschnitt 4 einen Großteil dieser Arbeit. Darin wird erklärt, wie verschiedene Aufgaben mit Hilfe der RFID-Technik in Kombination mit Industrieroboter und Greifer umgesetzt wurden. Das implementierte Gesamtsystem beherrscht nicht nur die typischen Funktionen jedes Teilsystems, sondern auch Kombinationen der Teilsystemfunktionen. So ist es möglich, dass die mit Transpondern bestückten Bausteine nach der Größe sortiert werden, nachdem deren Abmessungen in den Transpondern gespeichert wurde. Dabei kommen alle drei Teilsysteme – RFID, Roboter, Greifer – zum Einsatz. Weitere realisierte Funktionen sind beispielsweise

die Transponderortung im ein- und zweidimensionalen Bereich sowie das intelligente – vorsichtige, aber trotzdem schnelle – Greifen. Diese Funktionen beanspruchen nur einen Teil der Leistungsfähigkeit des Gesamtsystems und bilden eine gute Basis für künftige Anwendungen.

Auf der einen Seite können in künftigen Projekten die in dieser Arbeit beschriebenen, Funktionen verbessert werden – nicht nur was die Genauigkeit angeht. Zum Beispiel ist es vorstellbar, die Funktion des intelligenten Greifens nach eindimensionaler Bausteinortung um eine Dimension zu erweitern. Bisher ist dies lediglich möglich, wenn sich der Baustein auf einer vorher definierten Linie des Messbereiches befindet. Im Rahmen künftiger Projekte an der Hochschule Mittweida könnte ein Verfahren entwickelt werden, das einen Baustein an einer beliebigen Position des zweidimensionalen Messgebietes findet und zuverlässig greift. Analog ist es denkbar, weitere in dieser Arbeit implementierte Funktionen um eine Dimension zu erweitern. So könnte bei der Transponderortung mit der großen Rahmenantenne von der Ebene in den Raum übergegangen werden. Ferner kann die mit dieser Arbeit gelegte Basis genutzt werden, um die Kombination aus Roboter, Greifer und RFID-System grundlegend um ein weiteres Teilsystem zu erweitern. Es ist beispielsweise denkbar, eine optische Bildverarbeitung / ein Kamerasystem zu installieren. Das Potential dieser Kombination aus Roboter, Greifer und RFID-System ist jedenfalls noch längst nicht ausgeschöpft.

Sämtliche bereits umgesetzte Funktionen wurden in Form von Fotos und Videos dokumentiert. Diese sind, zusammen mit sämtlichen programmierten Matlab-Funktionen und Matlab-Scripten auf dem (in Anlage 2 beschriebenen) beiliegenden Datenträger zu finden. Zusammen mit dem installierten Versuchsstand können diese Materialien als Lehr- und Anschauungsmaterialien künftiger Studenten der Hochschule Mittweida genutzt werden.

Literatur & Quellen

- [ABB2013] ABB AB Robotics Products: Bedienungsanleitung Erste Schritte - IRC5 und RobotStudio, Västerås Schweden, 2013
- [ABB2016] ABB Ltd contact.center@de.abb.com: IRB 120 – Allgemein. URL: <http://www.abb.de/product/seitp327/2b2a762a99f114524825773f001c3cf1.aspx#!>, verfügbar am 19.12.2016
- [BioE2016] BioEnable Technologies Pvt. Ltd. Chipsilicon mayur@bioenabletech.com: 13.56MHz ISO15693 RFID iCode Smart Labels. URL: <http://shop.chipsilicon.com/13.56MHz-ISO15693-RFID-iCode-Smart-Labels-Pack-of-100->, verfügbar am 05.12.2016
- [Bral2017] Brall, Frank webmaster@roboternetz.de: Ethernet: Manchester-Code. URL: <http://rn-wissen.de/wiki/index.php?title=Manchester-Codierung>, , verfügbar am 26.01.2017
- [FEIG2009] FEIG ELECTRONIC GmbH: Montage Installation ID ISC.ANT310310-A HF Antenna, Weilburg-Waldhausen, 2009
- [FEIG2014] FEIG ELECTRONIC GmbH: Integrationsanleitung ID ISC.LRM1002 – E Long Range Reader Modul, Weilburg, 2014

- [FEIG2015] FEIG ELECTRONIC GmbH: Manual ID ISC.LR1002 Standard Reader, Weilburg-Waldhausen, 2015
- [Fink2016] Finkenzeller, Klaus: RFID Handbuch – 7.Auflage, München, Carl Hanser Verlag, 2015
- [GeKr2015] Gessler, Ralf; Krause, Thomas: Wireless-Netzwerke für den Nahbereich – 2. Auflage, Wiesbaden, Springer Fachmedien, 2015
- [ISO2006] customerservice@iso.org: ISO/IEC 15693-2:2006. URL: <https://www.iso.org/standard/39695.html>, verfügbar am 24.11.2016
- [ISO2009] Norm ISO/IEC 15693-3:2009: Identification cards - Contactless integrated circuit cards - Vicinity cards - Part 3: Anticollision and transmission protocol
- [Kern2011] Kern, Christian: RFID für Bibliotheken, Berlin, Springer, 2011
- [Koln2001] Kolnsberg, Stephan: Drahtlose Signal- und Energieübertragung von Hochfrequenztechnik in CMOS-Sensorsystemen, Duisburg, 2001
- [Lamm2017] Lammers, Sascha sascha-lammers@web.de: RFID. URL: <http://www.rfid-grundlagen.de>, verfügbar am 01.03.2017 Uhr

- [Lang2017a] Lang, Hans Werner lang@hs-flensburg.de: Bubblesort. URL: <http://www.iti.fh-flensburg.de/lang/algorithmen/sortieren/networks/bubble.htm>, verfügbar am 13.02.2017
- [Lang2017b] Lang, Hans Werner lang@hs-flensburg.de: Selectionsort. URL: <http://www.iti.fh-flensburg.de/lang/algorithmen/sortieren/networks/selection.htm>, verfügbar am 13.02.2017
- [Math2016] The MathWorks, Inc.: Die Sprache des technischen Rechnens. URL: <https://de.mathworks.com/products/matlab.html?requestedDomain=de.mathworks.com>, verfügbar am 19.12.2016
- [OvVo2005] Overmeyer, Ludger; Vogeler, Stefan: RFID: Grundlagen und Potentiale, Universität Hannover, Logistics Journal, 2005
- [Piet2014] Pietruszka, Wolf Dieter: Matlab und Simulink in der Ingenieurpraxis – 4. Auflage, Wiesbaden, Springer Fachmedien, 2014
- [PISc2016] Pläßmann, Wilfried; Schulz, Detlef: Handbuch Elektrotechnik, Wiesbaden, Springer Vieweg, 2016
- [Prec2007] Prechtel, Adalbert: Vorlesungen über die Grundlagen der Elektrotechnik: Band 2 – 2. Auflage, Wien, Springer, 2007

- [RFID2014] RFID Canada: Datenblatt ID ISC.ANT40/30 & ID ISC.ANT100/100 HF-Antennas for RFID proximity reader, Markham, Ontario Canada, 2014
- [RFID2016] webmaster@rfid-basis.de: Aufbau und Funktionsweise von RFID-Systemen. URL: <http://www.rfid-basis.de/funktionsweise.html>, verfügbar am 18.11.2016
- [Schn16] Schneider, Gerd: Auto-ID und Ortungs-lösungen aus einer Hand. URL: <http://www.solconia.de>, verfügbar am 05.12.2016
- [Schn2016] Schnabel, Patrick kontakt@das-ELKO.de: RFID - Radio Frequency Identification. URL: <http://www.elektronik-kompodium.de/sites/kom/0902021.htm>, verfügbar am 18.11.2016
- [Shar2008] Sharma, Sushil info@rfid-ready.de: RFID-Frequenzen. URL: <http://www.rfid-ready.de/rfid-frequenzen.html>, verfügbar am 24.10.2016
- [Stin2015] Stiny, Leonhard: Aktive elektronische Bauelemente – 2. Auflage, Wiesbaden, Springer Fachmedien, 2015
- [Tur2016a] Hans Turck GmbH & Co.KG: Datenblatt Mobiler Datenträger TW-R20-K2, Mülheim an der Ruhr, 2016

- [Tur2016b] Hans Turck GmbH & Co.KG: Datenblatt Mobiler Datenträger TW-L50-50-P-B128, Mülheim an der Ruhr, 2016
- [Tur2016c] Hans Turck GmbH & Co.KG: BL ident® RFID System Engineering User Manual, Mülheim an der Ruhr, 2016
- [Weis2014] Weiss Robotics: Montage- und Bedienungsanleitung Servoelektrischer Parallelgreifer WSG 50, 2014
- [WiME2006] Witschnig, Harald; Merlin, Erich: Über Geschichte, physikalische Grundlagen und Applikationen der RFID-Technologie. In: e&i Elektrotechnik und Informationstechnik, 2006, Heft 3, S.61 – 71. – ISSN: 0932-383X, Springer

Anlagen

Teil 1 - Sortieralgorithmen in Matlab.....II

Teil 2 - Beiliegender DatenträgerIV

Anlagen, Teil 1

Sortieralgorithmen in Matlab

```
%% Sortieralgorithmus Bubblesort
n=5; %Anzahl Bausteinpositionen = 5
z=0; %Zählvariable auf 0 setzen
while 1
    getauscht=0; %diese Variable prüft, ob eine Vertauschung stattfand
    for i=1:(n-1) %für die erste bis zur letzte Position
        if (breiten(i) > breiten(i+1))
            %wenn Element auf akt. Pos. größer als das nächster Pos.
            if breiten(i)==0
                %wenn Element auf akt. Pos. 0 ist (kein Baustein auf Pos.)
                tausche01(breiten,i,(i+1),rob,t);
                %tauscht Bausteine auf akt. und nächster Position
                z=z+1; %erhöht Zähler um 1
            else
                if breiten(i+1)==0
                    %wenn Element auf nächster Pos. 0 ist (kein Baustein auf
                    %Pos.)
                    tausche10(breiten,i,(i+1),rob,t);
                    %tauscht Bausteine auf akt. und nächster Position
                    z=z+1; %erhöht Zähler um 1
                else
                    tausche(breiten,i,(i+1),rob,t);
                    %tauscht Bausteine auf akt. und nächster Position
                    z=z+1; %erhöht Zähler um 1
                end
            end
        end
        getauscht=1; %vermerkt, dass getauscht wurde im Durchgang
        %Austausch von akt. und nächsten Element in Variable breiten
        x=breiten(i);
        breiten(i)=breiten(i+1);
        breiten(i+1)=x;
    end
end
if getauscht==0
    break; %beenden des Scripts, wenn kein Austausch im Durchgang
end
end
```

```

%% Sortieralgorithmus Selectionsort
n = 5; %Anzahl Bausteinpositionen = 5
z=0; %Zählvariable auf 0 setzen
for i = (1:n-1) %für die erste bis zur letzte Position
    minElem = breiten(i); %akt. Baustein wird als kleinster bestimmt
    minIndex = i; %akt. Pos. wird als die mit kleinsten Element bestimmt
    %weitere Positionen werden auf kleinere Elemente durchsucht
    for j = (i:n)
        if breiten(j) <= minElem
            minElem = breiten(j);
            minIndex = j;
        end
    end
    if i ~= minIndex %wenn größeres Element gefunden
        %wenn Element auf akt. Pos. 0 ist (kein Baustein auf Pos.)
        if breiten(i)==0
            tausche01(breiten,i,minIndex,rob,t);
            %tauscht Baustein auf akt. Pos. und Pos. mit kleinsten Baust.
            z=z+1; %erhöht Zähler um 1
        else
            %wenn minElem 0 ist (kein Baustein auf Pos.)
            if minElem==0
                tausche10(breiten,i,minIndex,rob,t);
                %tauscht Baust.auf akt.Pos. und Pos. mit kleinsten Baust.
                z=z+1; %erhöht Zähler um 1
            else
                tausche(breiten,i,minIndex,rob,t);
                %tauscht Baust.auf akt.Pos. und Pos. mit kleinsten Baust.
                z=z+1; %erhöht Zähler um 1
            end
        end
        breiten([minIndex i]) = breiten([i minIndex]);
        %Austausch von akt. und nächstkleinsten Element in Var. breiten
    end
end

end

%% Sortieralgorithmus Selectionsort verbessert
for Akt_feld=1:(Anzahl-1)
    %ist aktuelles Feld kleiner als alle nachfolgenden?
    if breiten(Akt_feld)~=min(breiten(Akt_feld:Anzahl))
        %aktuelles Feld beinhaltet nicht die nächst-kleinere Zahl
        %% Schritt 1: tausche i mit Leerfeld
        %finde Leerfeld (größte Zahl)
        Leerfeld=find(breiten==max(breiten));
        %falls Leerfeld nicht aktuelles Feld ist
        if Akt_feld~=Leerfeld
            %tausche Leerfeld mit akt. Feld (aktuelles Feld leermachen)
            breiten=tausche_rob(breiten,Akt_feld,min(Leerfeld),rob,t);
            Zaehler=anzeigen(Zaehler,Akt_feld,min(Leerfeld),breiten);
        end
        %% Schritt 2: tausche min mit Leerfeld
        %finde nächst-kleinere Zahl
        Minimum=find(breiten==min(breiten(Akt_feld:Anzahl)));
        %tausche aktuelles Feld(Leerfeld) mit nächst-kleinerer Zahl
        breiten=tausche_rob(breiten,max(Minimum),Akt_feld,rob,t);
        Zaehler=anzeigen(Zaehler,max(Minimum),Akt_feld,breiten);
    end
end
end

```

Anlagen, Teil 2

Beiliegender Datenträger

Ordner	Inhalt
Bilder	Fotos des Versuchsstandes und einzelner Komponenten
Masterarbeit	Digitale Version dieser Masterarbeit
Programmierungen	Matlab-Funktionen und -Scripte
Quellen	Verwendete Literaturquellen
Videos	Videos des Versuchsstandes während der Programmabläufe

Selbstständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Literatur und Hilfsmittel angefertigt habe.

Stellen, die wörtlich oder sinngemäß aus Quellen entnommen wurden, sind als solche kenntlich gemacht.

Diese Arbeit wurde in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegt.

Mittweida, den

.....

Richard Schneider